

日本国特許庁  
JAPAN PATENT OFFICE

31. 3. 2004

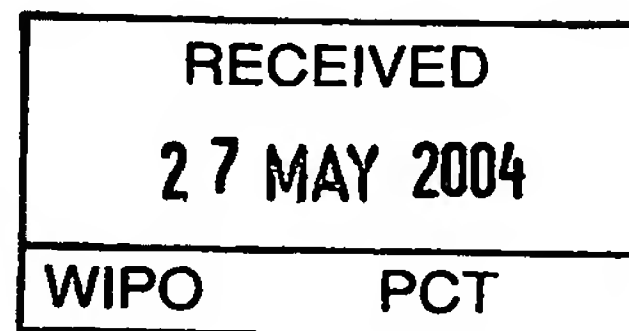
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2003年 3月31日

出願番号  
Application Number: 特願2003-093922  
[ST. 10/C]: [JP 2003-093922]

出願人  
Applicant(s): 財団法人北九州産業学術推進機構

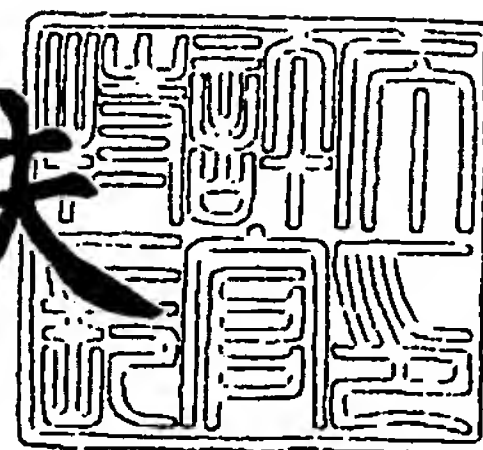


**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

2004年 5月13日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



【書類名】 特許願  
【整理番号】 J030128IK0  
【あて先】 特許庁長官殿  
【国際特許分類】 H03K 19/173  
G06F 7/00

## 【発明者】

【住所又は居所】 福岡県福岡市中央区今川 1 丁目 1 0 - 4 3 - 4 0 3

【氏名】 笹尾 勤

## 【発明者】

【住所又は居所】 神奈川県足柄下郡箱根町強羅 1 3 2 0 - 1 2 3 1

【氏名】 井口 幸洋

## 【特許出願人】

【識別番号】 502428467

【氏名又は名称】 笹尾 勤

## 【代理人】

【識別番号】 100099508

## 【弁理士】

【氏名又は名称】 加藤 久

【電話番号】 092-413-5378

## 【手数料の表示】

【予納台帳番号】 037590

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ルックアップテーブル・カスケード論理回路

【特許請求の範囲】

【請求項 1】 複数の入力変数を有する目的論理関数を関数分解して得られる複数の分解関数を、逐次計算することにより、目的論理関数の演算を行うルックアップテーブル・カスケード論理回路であって、

前記各分解関数をルックアップテーブルとして記憶する、直列に配列された複数の論理関数メモリと、

前段の前記論理関数メモリの複数個の出力及び前記入力変数の入力のうち、後段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶する接続メモリと、

前記接続メモリより出力される接続情報に従い、前段の前記論理関数メモリの出力又は前記入力変数の入力と後段の前記論理関数メモリの入力とを選択的に接続する接続回路と、

を備えていることを特徴とするルックアップテーブル・カスケード論理回路。

【請求項 2】 前記各論理関数メモリについて、当該論理関数メモリ内の入出力が可能なページを選択的に切り換えるページ切換手段を備えたことを特徴とする請求項 1 記載のルックアップテーブル・カスケード論理回路。

【請求項 3】 前記接続回路は、クロスバススイッチ、マルチプレクサ、若しくはシフタ回路により構成されていることを特徴とする請求項 1 又は 2 記載のルックアップテーブル・カスケード論理回路。

【請求項 4】 前記接続回路は、前段の前記論理関数メモリの出力を、論理関数の演算結果を出力する出力ラインに接続することが可能であることを特徴とする請求項 1 乃至 3 の何れかーに記載のルックアップテーブル・カスケード論理回路。

【請求項 5】 前記各接続回路の前記出力ラインからの出力又は最終段の前記論理関数メモリの出力のうちの何れかを選択し出力する第 1 の出力選択回路を備えていることを特徴とする請求項 4 記載のルックアップテーブル・カスケード論理回路。

【請求項 6】 最終段の前記論理関数メモリの出力を、  
w 入力 1 出力 ( $w \geq 2$ ) のマルチプレクサが多段にカスケード接続され、各段において、出力を取り出すことを可能とした回路を通して出力する第 2 の出力選択回路を備えていること  
を特徴とする請求項 1 乃至 5 の何れかーに記載のルックアップテーブル・カスケード論理回路。

【請求項 7】 前記第 1 の出力選択回路が選択した出力を、  
w 入力 1 出力 ( $w \geq 2$ ) のマルチプレクサが多段にカスケード接続され、各段において、出力を取り出すことを可能とした回路を通して出力する第 2 の出力選択回路を備えていること  
を特徴とする請求項 5 記載のルックアップテーブル・カスケード論理回路。

【請求項 8】 最初の段以外の前記各論理関数メモリにおいては、その入力のうちの少なくとも 1 つには、前記接続回路を介することなく前記入力変数を直接入力することを特徴とする請求項 1 乃至 7 記載のルックアップテーブル・カスケード論理回路。

【請求項 9】 前記各入力変数を記憶する入力変数レジスタと、  
前記入力変数レジスタから出力される入力変数のうちの一部を選択し、前記論理関数メモリ又は前記接続回路に入力する入力選択回路と、  
を備えていることを特徴とする請求項 1 乃至 8 の何れかーに記載のルックアップテーブル・カスケード論理回路。

【請求項 10】 前記入力選択回路は、シフタ回路により構成されていることを特徴とする請求項 9 記載のルックアップテーブル・カスケード論理回路。

【請求項 11】 前記入力選択回路は、前記各接続回路又は前記各論理関数メモリのそれぞれに対して設けられていることを特徴とする請求項 9 又は 10 に記載のルックアップテーブル・カスケード論理回路。

【請求項 12】 最初の段以外の前記各論理関数メモリは、前段の前記論理関数メモリの出力の一部が、前記接続回路を介することなく後段の前記論理関数メモリの入力の一部に接続されていることを特徴とする請求項 1 乃至 11 の何れかーに記載のルックアップテーブル・カスケード論理回路。

## 【発明の詳細な説明】

## 【0 0 0 1】

## 【発明の属する技術分野】

本発明は、プログラム可能な論理回路に係り、特に、目的論理関数を関数分解して得られる分解関数を、逐次計算することにより、目的論理関数の演算を行うルックアップテーブル・カスケード論理回路に関する。

## 【0 0 0 2】

## 【従来の技術】

近年、論理回路の設計においては、マトリックス状に配列された複数の論理セルの内容と論理セル間の配線の接続とをメモリの書き換えによって変更することが可能なフィールド・プログラマブル・ゲートアレイ（Field Programmable Gate Array。以下、「FPGA」という。）が広く用いられるようになっている。FPGAは、プログラムに従ってソフトウェア的に演算を行うマイクロ・プロセッサ（以下、「MPU」という。）とは異なり、ハードウェア的に演算を行うことから、論理関数の演算実行速度が速いという特徴を有する。

## 【0 0 0 3】

その一方、FPGAは、物理的な配線の引き回しをプログラムにより変更することから、配線遅延を考慮した設計が必要とされる。そのため、論理回路の設計に長時間を要する。また、各論理セル間の配線の引き回し方により配線遅延時間が変わる。そのため、設計時において論理回路の演算時間の予測が困難である。

## 【0 0 0 4】

そこで、上記FPGAの欠点をカバーするプログラム可能な論理回路として、ルックアップテーブル・カスケード論理回路が提案されている（非特許文献1，2参照）。以下、ルックアップテーブル・カスケード論理回路について簡単に説明する。

## 【0 0 0 5】

図22はルックアップテーブル・カスケード論理回路の原理を表す図である。

ルックアップテーブル・カスケード論理回路において論理関数の演算を行う場合、まず、演算を行う論理関数（目的論理関数） $f(X)$ を $p$ 個の分解関数  $\{f_i(X_i)$



;  $i=0, \dots, p-1$  } に関数分解する。ここで、 $X=(x_0, \dots, x_{n-1})$  は入力変数を表す。また、入力変数の集合を  $\{X\}$  で表すと、 $\{X\} = \{X_0\} \cup \dots \cup \{X_{p-1}\}$ ,  $\{X_i\} \cap \{X_j\} = \emptyset$  ( $i \neq j$ ;  $i, j \in \{0, \dots, p-1\}$ ) である。各分解関数  $f_i$  の出力変数 (一般に、ベクトル) を  $Y_{i+1}$  で表す。以下、 $X, Y$  の変数の個数を  $|X|$ ,  $|Y|$  のように表すこととし、特に、 $|X| = n$ ,  $|X_i| = n_i$ ,  $|Y_i| = u_i$  と表記する。

#### 【0 0 0 6】

任意の目的論理関数  $f$  は、 $n_0=s$  ( $s$  は分解関数  $f_0$  の入力変数の数)、 $n_i=s-u_i$  ( $0 < i < p-1$ ) ,  $n_{p-1}=s-u_{p-1}-t$  ( $0 \leq t \leq s-2$ ) となるように関数分解することが可能である。そこで、目的論理関数  $f$  を  $p-1$  個の  $s$  入力の分解関数  $f_j$  ( $j \in \{0, \dots, p-2\}$ ) と 1 個の  $s-t$  入力の分解関数  $f_{p-1}$  とに分解する。そして、各分解関数  $f_k$  ( $k \in \{0, \dots, p-1\}$ ) を真理値表により表し、これをルックアップテーブルとする。

#### 【0 0 0 7】

図 2 2 において、 $p$  個のルックアップテーブル  $LUT_0 \sim LUT_{p-1}$  は、それぞれ、分解関数  $\{f_i; i=0, \dots, p-1\}$  を真理値表により表現したものである。それぞれのルックアップテーブル  $LUT_k$  は、 $s$  入力  $u_{k+1}$  出力の論理関数メモリを用いて実現することができる。このような論理関数メモリを、図 2 2 に示すようにカスケード接続することにより、ルックアップテーブル・カスケード論理回路が実現される。

#### 【0 0 0 8】

ルックアップテーブル・カスケード論理回路では、目的論理関数  $f$  の入出力変数の個数が多い場合には、多数のルックアップテーブルをカスケード接続する必要があるため、最適化された F P G A に比べれば演算速度は遅くなる場合もある。しかし、ルックアップテーブル・カスケード論理回路では、各分解関数の演算について、論理関数メモリを用いてハードウェア的に行うことから、M P U に比べると演算速度を高速化することができる。そして、ルックアップテーブルの段数によって、一義的に演算速度が定まることから、論理回路の設計時に目的論理関数の演算時間を予測することが容易である。また、互いに隣接する論理関数メモリ間でのみ配線を行えばよいことから、論理回路の設計時には配線遅延の影響を考慮する必要がなく、F P G A に比べると論理回路設計が格段に容易になる

## 【 0 0 0 9 】

## 【非特許文献 1】

笹尾勤, 松浦宗寛, 井口幸洋, “多出力関数のカスケード実現と再構成可能ハードウェアによる実現”, 電子情報通信学会FTS研究会, FTS2001-8, pp. 57-64, 三重大学(2001-04).

## 【非特許文献 2】

T. Sasao, M. Matsuura, and Y. Iguchi, "A cascade realization of multiple-output function for reconfigurable hardware," International Workshop on Logic and Synthesis (IWLS01), Lake Tahoe, CA, June 12-15, 2001. pp.225-230.

## 【 0 0 1 0 】

## 【発明が解決しようとする課題】

しかしながら、実際に図 2 2 に示したようなルックアップテーブル・カスケード論理回路により目的論理関数の演算を行う場合には、目的論理関数を関数分解して得られる分解関数のルックアップテーブル間のレイル数（二つのルックアップテーブル間の線数（中間変数の数））や各ルックアップテーブルの入力変数の個数が目的論理関数により異なる。そのため、種々の目的論理関数をルックアップテーブル・カスケード論理回路で実現しようとした場合、入力変数の入力ラインの数及びレイル数が異なる種々のルックアップテーブル・カスケードが必要となり、実用性に欠ける。

## 【 0 0 1 1 】

そこで、多様な目的論理関数に対応可能なルックアップテーブル・カスケード論理回路とするために、各論理関数メモリの入力のビット数に余裕をもたせ、あらかじめビット数を大きめにとっておくことが考えられる。しかし、論理関数メモリのメモリ容量は、論理関数メモリの入力のビット数の増加に伴い増大する。そして、使用されないメモリ領域が増大し、無駄が多くなる。また、メモリ容量の増大に伴いメモリのレイアウト面積が増大するため、回路の高集積化が困難となり、回路消費電力も大きくなる。

## 【 0 0 1 2 】

そこで、本発明の目的は、目的論理関数に応じて、各論理関数メモリ間の入力変数の入力ライン数とレイル数とを柔軟に変化させることが可能で、論理関数メモリの入力数、延いては論理関数メモリのメモリ容量を必要最小限に抑えて設計することが可能なルックアップテーブル・カスケード論理回路を提供することにある。

## 【 0 0 1 3 】

## 【課題を解決するための手段】

本発明のルックアップテーブル・カスケード論理回路の第 1 の構成は、複数の入力変数を有する目的論理関数を関数分解して得られる複数の分解関数を、逐次計算することにより、目的論理関数の演算を行うルックアップテーブル・カスケード論理回路において、前記各分解関数をルックアップテーブルとして記憶する、直列に配列された複数の論理関数メモリと、前段の前記論理関数メモリの複数個の出力及び前記入力変数の入力のうち、後段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶する接続メモリと、前記接続メモリより出力される接続情報に従い、前段の前記論理関数メモリの出力又は前記入力変数の入力と後段の前記論理関数メモリの入力とを選択的に接続する接続回路と、を備えていることを特徴とする。

## 【 0 0 1 4 】

この構成によれば、各段の論理関数メモリの入力に対して、前段の論理関数メモリの出力及び入力変数の入力は、接続情報に従って接続回路を介して後段の論理関数メモリの入力に選択的に接続される。従って、接続メモリ内の接続情報を目的論理関数に合わせて書き換えれば、後段の論理関数メモリに接続する前段の論理関数メモリの出力の個数と、後段の論理関数メモリに接続する入力変数の入力の個数を、目的論理関数に応じて自在に変更することが可能となる。すなわち、目的論理関数に応じて、各分解関数のレイル数と外部からの入力変数の入力数との割合を任意に変更することが可能となる。そのため、1つのルックアップテーブル・カスケード論理回路で実現可能な目的論理関数の自由度が大きくなる。その結果、各論理関数メモリの入力の配線数を減らすことが可能となり、回路を



小型化することができる。また、各分解関数のレイル数と外部からの入力変数の入力数との割合を、分解関数に応じて最適化することができるため、論理関数メモリの入力線の配分、メモリの使用領域に無駄をなくすることができる。

#### 【 0 0 1 5 】

本発明の第 2 の構成は、前記第 1 の構成において、前記各論理関数メモリについて、当該論理関数メモリ内の入出力が可能なページを選択的に切り換えるページ切換手段を備えたことを特徴とする。

#### 【 0 0 1 6 】

この構成によれば、目的論理関数に応じて論理関数メモリのページを指定して、指定されたページに当該目的論理関数の分解関数のルックアップテーブルを格納する。これにより、1つのルックアップテーブル・カスケード論理回路を用いて複数の目的論理関数の論理回路を設計することができる。

#### 【 0 0 1 7 】

本発明の第 3 の構成は、前記第 1 又は 2 の構成において、前記接続回路は、クロスバススイッチ、マルチプレクサ、若しくはシフタ回路により構成されていることを特徴とする。

#### 【 0 0 1 8 】

このように、接続回路をクロスバススイッチ、マルチプレクサ、若しくはシフタ回路とすることで、簡単な回路により接続回路を実現することができる。

#### 【 0 0 1 9 】

特に、回路の演算速度の高速化が要求される場合には、接続回路としてクロスバススイッチを使用することが好ましい。1本の配線上の切換スイッチの個数が少なく、切換スイッチにおける遅延を極力減らすことが可能だからである。

#### 【 0 0 2 0 】

また、回路の特に高集積化・低消費電力化が要求される場合には、接続回路としてシフタ回路を使用することが好ましい。接続関係の自由度は少ないが、接続メモリに必要とされる、接続回路の接続切換を制御するための制御線の数をも最小限とすることが可能となり、構成素子も少なく回路のレイアウト面積を小さくできるからである。

## 【 0 0 2 1 】

本発明の第 4 の構成は、前記第 1 乃至 3 の何れか一の構成において、前記接続回路は、前段の前記論理関数メモリの出力を、論理関数の演算結果を出力する出力ラインに接続することが可能であることを特徴とする。

## 【 0 0 2 2 】

この構成により、カスケード接続された論理関数メモリうちの任意のものから、その出力を取り出すことができる。従って、目的論理関数の分解関数の段数が少ない場合には、カスケードの途中の論理関数メモリの出力を出力変数として取り出すことで、演算処理を早期に終了させて演算速度を高速化することができる。

## 【 0 0 2 3 】

また、多変数出力論理関数の演算を行う場合には、ルックアップテーブル・カスケードの途中で確定した演算結果を、後段のルックアップテーブル・カスケードに入力することなく、途中で出力させることにより、後段のルックアップテーブル・カスケードのレイル数を削減することができる。これにより、演算処理速度を上げることができるとともに、後段のルックアップテーブル・カスケードに必要とされるメモリ容量の大幅な削減が可能となる。

## 【 0 0 2 4 】

本発明の第 5 の構成は、前記第 4 の構成において、前記各接続回路の前記出力ラインからの出力又は最終段の前記論理関数メモリの出力のうちの何れかを選択し出力する第 1 の出力選択回路を備えていることを特徴とする。

## 【 0 0 2 5 】

この構成により、出力回路により、各接続回路の前記出力ラインからの出力又は最終段の論理関数メモリの出力のうち、目的論理関数の演算結果が出力されるものを選択して出力させることができる。

## 【 0 0 2 6 】

本発明の第 6 の構成は、前記第 1 乃至 5 の何れか一の構成において、前記最終段の前記論理関数メモリの出力を、 $w$ 入力 1 出力 ( $w \geq 2$ ) のマルチプレクサが多段にカスケード接続され、各段において、出力を取り出すことを可能とした

回路を通して出力する第2の出力選択回路を備えていることを特徴とする。

#### 【0027】

この構成により、第2の出力選択回路によって最終段の分解関数の出力変数を選択し、出力変数の個数を $1/2$ 以下にして出力することが可能となる。すなわち、最終段の分解関数の出力変数の個数が、最終段の論理関数メモリの出力数の $1/2$ 以下の場合、入力変数のうちの一部のものの値に基づいて、最終段の分解関数の出力変数を選択することができ、第2の出力選択回路において更に論理演算を行うことができる。従って、各論理関数メモリに入力する入力変数の個数を削減することができるため、論理関数メモリのメモリ使用効率が向上する。また、ルックアップテーブル・カスケード論理回路全体で演算が可能な論理関数の入力変数の個数に関する制限数を広げることが可能となる。

#### 【0028】

本発明の第7の構成は、前記第5の構成において、前記第1の出力選択回路が選択した出力を、 $w$ 入力1出力 ( $w \geq 2$ ) のマルチプレクサが多段にカスケード接続され、各段において、出力を取り出すことを可能とした回路を通して出力する第2の出力選択回路を備えていることを特徴とする。

#### 【0029】

この構成により、第2の出力選択回路によって各段の分解関数の出力変数を選択し、出力変数の個数を $1/2$ 以下にして出力することが可能となる。すなわち、各段の分解関数の出力変数の個数が、各段の論理関数メモリの出力数の $1/2$ 以下の場合、入力変数のうちの一部のものの値に基づいて、各段の分解関数の出力変数を選択することができ、第2の出力選択回路において更に論理演算を行うことができる。従って、各論理関数メモリに入力する入力変数の個数を削減することができるため、論理関数メモリのメモリ使用効率が向上する。また、ルックアップテーブル・カスケード論理回路全体で演算が可能な論理関数の入力変数の個数に関する制限数を広げることが可能となる。

#### 【0030】

本発明の第8の構成は、前記第1乃至7の構成において、最初の段以外の前記各論理関数メモリにおいては、その入力のうちの少なくとも1つには、前記接続

回路を介すことなく前記入力変数を直接入力することを特徴とする。

【0031】

最初の段の論理関数メモリについては、総ての入力に対して入力変数が入力される。一方、最初の段以外の各論理関数メモリにおいては、その入力の一部には前段の論理関数メモリの出力が入力され、他の入力には、入力変数が入力される。しかしながら、最初の段以外の各論理関数メモリには、最低限 1 つの入力変数が入力される。従って、最初の段以外の各論理関数メモリの入力のうちの少なくとも 1 つに、接続回路を介すことなく入力変数を直接入力することにより、接続回路の入力ラインの数を減らすことができる。これにより、接続回路及び接続メモリの出力の配線数を減らすことができるため、回路を小型化することが可能となる。

【0032】

本発明の第 9 の構成は、前記第 1 乃至 8 の何れか一の構成において、前記各入力変数を記憶する入力変数レジスタと、前記入力変数レジスタから出力される入力変数のうちの一部を選択し、前記論理関数メモリ又は前記接続回路に入力する入力選択回路と、を備えていることを特徴とする。

【0033】

この構成によれば、入力選択回路は、入力変数レジスタに記憶された入力変数のうち、少なくとも実行中の分解関数の演算で使用する入力変数を含む変数を選択して出力する。従って、接続回路に入力される入力変数の個数を減らすことができる。そのため、接続回路及び接続メモリの出力の配線数を減らすことができるため、回路を小型化することが可能となる。

【0034】

本発明の第 10 の構成は、前記第 9 の構成において、前記入力選択回路は、シフト回路により構成されていることを特徴とする。

【0035】

この構成により、入力変数の選択は、入力選択回路であるシフト回路と接続回路とにより行われる。すなわち、シフト回路によって入力レジスタに記憶された入力変数のうち、先頭から k ビットだけシフトさせた入力変数から所定のビット



分の入力変数が選択されて接続回路又は論理関数メモリに出力される。選択回路は、入力された入力変数のうち、必要なものを選択して論理関数メモリに入力する。従って、簡易な回路によって入力選択回路を構成することができる。また、入力選択回路における入力変数の選択も、入力変数のシフトビット数を指定するだけでよい。ため、入力選択回路の制御ラインが少なく、配線面積を減らすことができる。

#### 【 0 0 3 6 】

本発明の第 1 1 の構成は、前記第 9 又は 1 0 の構成において、前記入力選択回路は、前記各接続回路又は前記各論理関数メモリのそれぞれに対して設けられていることを特徴とする。

#### 【 0 0 3 7 】

この構成により、各接続回路及び各論理関数メモリに対して入力する入力変数を、各入力選択回路により個別に選択することができる。従って、各論理関数メモリの演算を同時に行うことが可能であり、パイプライン処理が可能となる。

#### 【 0 0 3 8 】

本発明の第 1 2 の構成は、前記第 1 乃至 1 1 の構成において、最初の段以外の前記各論理関数メモリは、前段の前記論理関数メモリの出力の一部が、前記接続回路を介することなく後段の前記論理関数メモリの入力の一部に接続されていることを特徴とする。

#### 【 0 0 3 9 】

最初の段以外の各論理関数メモリにおいては、その入力の一部には前段の論理関数メモリの出力が入力され、他の入力には、入力変数が入力される。しかしながら、最初の段以外の各論理関数メモリには、最低 1 つは前段の論理関数メモリの出力が入力される。従って、最初の段以外の各論理関数メモリの入力の一部を、接続回路を介することなく前段の論理関数メモリの出力を直接入力することにより、接続回路の入力ラインの数を減らすことができる。これにより、接続回路及び接続メモリの出力の配線数を減らすことができるため、回路を小型化することが可能となる。

#### 【 0 0 4 0 】



**【発明の実施の形態】**

以下、本発明の一実施形態について、図面を参照しながら説明する。

**【0041】**

(実施形態1)

図1は本発明の実施形態1に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

**【0042】**

本発明の実施形態1に係るルックアップテーブル・カスケード論理回路は、入力変数レジスタ1、入力変数選択回路2-0～2-3、入力選択メモリ3-0～3-3、論理関数メモリ4-0～4-3、接続回路5-1～5-3、接続メモリ6-1～6-3、及び演算制御部10を備えている。

**【0043】**

入力変数レジスタ1は、ルックアップテーブル・カスケード論理回路によって演算を行う論理関数（目的論理関数） $f(X)$ の演算に用いられる $n$ 個の入力変数 $X=(x_1, \dots, x_n)$ （但し、 $n \in \text{自然数}$ ）を記憶する。論理関数メモリ4-0～4-3は、目的論理関数 $f$ を関数分解して得られる分解関数 $\{f_i; i \in \{0, 1, 2, 3\}\}$ の真理値表を、ルックアップテーブルとして記憶する。尚、本実施形態においては、一例として、論理関数メモリ4- $i$ （ $i=0, 1, 2, 3$ ）の段数を4段としているが、一般にはこの段数は任意に設定することができる。これらの論理関数メモリ4-0～4-3は、直列に配列され、それぞれ接続回路5-1～5-3を介して直列に接続されている。

**【0044】**

入力変数選択回路2-0～2-3は、入力変数レジスタ1から出力される $n$ 個の入力変数 $X=(x_1, \dots, x_n)$ のうちから、それぞれ、各段の論理関数メモリ4-0～4-3に記憶された分解関数 $\{f_i; i \in \{0, 1, 2, 3\}\}$ の真理値表の入力変数 $X_i$ （ $i \in \{0, 1, 2, 3\}$ ）を選択し、論理関数メモリ4-0又は接続回路5-1～5-3へ出力する。入力選択メモリ3-0～3-3は、それぞれ、入力選択回路2-0～2-3の入力変数の選択に関する情報（以下、「入力選択情報」という。）が記憶されている。入力変数選択回路2-0～2-3は、各入力選択メモリ3-0～3-3から出力さ

れる入力選択信号に基づいて、入力変数の選択の切り換えを行う。

#### 【0045】

各接続回路 5-i ( $i \in \{1, 2, 3\}$ ) は、論理関数メモリ 4-(i-1) 及び入力選択回路 2-i より入力される、中間変数  $Y_i$  及び入力変数  $X_i$  を、順序を変えて、後段の論理関数メモリ 4-i 及び外部出力ライン 7-i へ接続する。各接続メモリ 6-i は、各接続回路 5-i の接続関係に関する情報（以下、「接続情報」という。）を記憶している。接続回路 5-i は、接続メモリ 6-i から出力される接続情報信号に基づき、接続関係の切り換えを行う。

#### 【0046】

演算制御部 10 は、ルックアップテーブル・カスケード論理回路全体の演算処理の制御を行う。

#### 【0047】

図 2 は図 1 の入力変数選択回路 2-i ( $i \in \{0, 1, 2, 3\}$ ) の構成を表す回路図である。尚、この回路図は、動作原理を説明するために、簡略化して記載している。

#### 【0048】

本実施形態の入力変数選択回路 2-i ( $i \in \{0, 1, 2, 3\}$ ) は、図 2 のようなシフト回路により構成されている。尚、図 2 では、説明の都合上、17 入力 8 出力の入力変数選択回路 2-i の例を示しているが、入出力の数は、これに限られるものではない。

#### 【0049】

入力変数選択回路 2-i ( $i \in \{0, 1, 2, 3\}$ ) の入力端子 in(00) ~ in(16) は、入力変数レジスタ 1 の出力端子に接続されている。従って、入力端子 in(00) ~ in(16) からは入力変数  $X = (x_1, \dots, x_n)$  が入力される。また、入力変数選択回路 2-0 の出力端子 out(00) ~ out(07) は、論理関数メモリ 4-0 の入力に接続されている。入力変数選択回路 2-i ( $i \in \{1, 2, 3\}$ ) の出力端子 out(00) ~ out(07) は、接続回路 5-i の入力側端子の一部に接続されている。

#### 【0050】

入力変数選択回路 2-i ( $i \in \{0, 1, 2, 3\}$ ) は、入力側から、8 ビットシフト回路 11-3、4 ビットシフト回路 11-2、2 ビットシフト回路 11-1、及び 1 ビットシフト回路 11-0 により構成されている。

ットシフト回路 11-0 が直列に接続されている。これにより、入力変数選択回路 2-i ( $i \in \{0, 1, 2, 3\}$ ) は、入力端子 in(00)~in(16) の内容を 0~15 ビットの範囲でシフトさせて、連続する 8 ビット分を出力端子 out(00)~out(07) に出力させることができる。

#### 【0051】

各シフト回路 11-0 ~ 11-3 には、1 ビットの制御ライン shf0~shf3 が接続されている。制御ライン shf j ( $j \in \{0, 1, 2, 3\}$ ) が 0 のときは、シフト回路 11-i は接続のシフトを行わず、制御ライン shf j ( $j \in \{0, 1, 2, 3\}$ ) が 1 のときは、シフト回路 11-i は接続のシフトを行う。尚、入力変数選択回路 2-i ( $i \in \{0, 1, 2, 3\}$ ) の各制御ライン shf0~shf3 は、入力選択メモリ 3-i の出力に接続されている。すなわち、入力選択メモリ 3-0 から読み出された 4 ビットのメモリの内容（すなわち、入力選択情報）が、選択制御信号としてそのまま制御ライン shf0~shf3 に出力され、入力選択回路 2-i ( $i \in \{0, 1, 2, 3\}$ ) における入力変数のシフト量が設定される。

#### 【0052】

尚、図 2 では、簡略化のために省略しているが、シフタ回路 11-0 ~ 11-3 内の各パストランジスタにより入力変数 X の信号振幅が減衰する。従って、入力変数 X の信号振幅を補償するために、シフタ回路の数段ごとに増幅器（バッファ）を挿入しておくことが必要となる。

#### 【0053】

図 3 は図 1 の論理関数メモリ 4-i ( $i \in \{0, 1, 2, 3\}$ 、以下、符号 4-0 ~ 4-3 をまとめていう場合には、符号 4 で表す。) の構成を表すブロック図である。

#### 【0054】

論理関数メモリ 4 は、その内部に、m ページからなるメモリアレイ FM<sub>0</sub>~FM<sub>m-1</sub> を備えている。これらのメモリアレイには、分解関数  $f_i(j)$  ( $i \in \{0, 1, 2, 3\}$ ,  $j \in \{0, \dots, m-1\}$ ) の真理値表がルックアップテーブル LUT(j) として格納されている。

#### 【0055】

尚、ここで、添字「(j)」 ( $j \in \{0, \dots, m-1\}$ ) は、j 番目の目的論理関数 f(j) を表す。本実施形態におけるルックアップテーブル・カスケード論理回路では、複

数の目的論理関数 $f(j)$ の分解関数 $\{f_i(j)\}$ を、それぞれの論理関数メモリ4の第 $j$ ページに記憶させておき、使用目的に応じて、ページを切り換えて、目的論理関数を選択することが可能とされている。

#### 【0056】

また、論理関数メモリ4- $i$  ( $i \in \{0, 1, 2, 3\}$ ) は、ページデコーダ15及びアドレスデコーダ16を備えている。ページデコーダ15は、演算制御部10から入力されるページ選択番号 $p_k$ に基づいて、メモリアレイ $FM_k$ のメモリアクセスを可能とする。アドレスデコーダ16は、接続回路から $N_{in}$ ビットの入力ライン17を介して入力される変数 $(X_i, Y_i)$  ( $i \in \{0, 1, 2, 3\}$ 、但し、 $i=0$ のとき、 $Y_i = \phi$ 。)に基づき、メモリアレイ $FM_k$ 内のメモリ内容を読み出すアドレスを選択する。この選択されたアドレスのメモリセルには、分解関数 $f_k(j)$ の真理値 $f_k(j)(X_k, Y_k)$ が格納されている。メモリアレイ $FM_k$ は、ページデコーダ15及びアドレスデコーダ16により、メモリセル内のアドレスが指定されると、当該メモリセル内に記憶されているデータを、中間変数 $Y_{k+1} = (y_{k+1,1}, \dots, y_{k+1,|Y_{k+1}|})$ として出力ライン18に出力する。

#### 【0057】

図4は図1の接続回路5- $i$  ( $i \in \{1, 2, 3\}$ 、以下、符号5-1～5-3をまとめていう場合には、符号5で表す。)の周辺の回路ブロック図である。尚、この回路図は、動作原理を説明するために、簡略化して記載している。

#### 【0058】

本実施形態における接続回路5は、入力ライン $i_0 \sim i_{15}$ からの入力信号を循環的に任意のビット数だけシフトさせて出力ライン $o_1 \sim o_{15}$ に出力するシフト回路により構成されている。尚、図4においては、説明の都合上、一例として、16ビット入力16ビット出力の接続回路を記載しているが、入出力のビット数は、特にこれに限定されない。

#### 【0059】

接続回路5は、入力ライン $i_0 \sim i_{15}$ の側から、8ビットシフト回路20-3、4ビットシフト回路20-2、2ビットシフト回路20-1、1ビットシフト回路20-0が直列に接続された構成とされている。各シフト回路20- $i$  ( $i \in \{0, 1, 2,$

3}) のオン／オフは、接続メモリ 6 (符号 6-1 ~ 6-3 をまとめて符号 6 で表す。以下同じ。) の出力に接続された制御ライン  $s_j$  ( $j \in \{0, 1, 2, 3\}$ ) により制御される。

#### 【0060】

接続回路 5-k ( $k \in \{1, 2, 3\}$ ) の入力ライン  $i_0 \sim i_{15}$  のうち、 $i_0 \sim i_7$  は前段の論理関数メモリ 4-(k-1) の出力側に接続されており、 $i_8 \sim i_{15}$  は入力変数選択回路 2-k の出力側に接続されている。また、接続回路 5-k の出力ライン  $o_0 \sim o_{15}$  のうち、 $o_0 \sim o_7$  は、後段の論理関数メモリ 4-k の入力側に接続されており、 $o_8 \sim o_{15}$  は外部出力ライン 7-k となっている。

#### 【0061】

各シフト回路 20-0 ~ 20-3 の接続関係のシフトがないときには、入力ライン  $i_0 \sim i_{15}$  は、それぞれ出力ライン  $o_0 \sim o_{15}$  に接続される。これにより、前段の論理関数メモリ 4-(k-1) の出力である中間変数  $Y_k$  は、総て後段の論理関数メモリ 4-k に入力される。

#### 【0062】

一方、シフト回路 20-3 のみが接続関係を 8 ビットシフトさせた場合には、入力ライン  $i_0 \sim i_7$  が、それぞれ、出力ライン  $o_8 \sim o_{15}$  に接続され、入力ライン  $i_8 \sim i_{15}$  が、それぞれ、出力ライン  $o_0 \sim o_7$  に接続される。これにより、前段の論理関数メモリ 4-(k-1) の出力である中間変数  $Y_k$  は、総て外部出力ライン 7-k に出力される。また、入力変数選択回路 2-k から出力される入力変数  $X_k$  は、総て後段の論理関数メモリ 4-k に入力される。

#### 【0063】

また、各シフト回路 20-0 ~ 20-2 により、接続関係を  $j$  ビット ( $1 \leq j \leq 7$ ) だけシフトさせた場合には、入力ライン  $i_{15-j+1} \sim i_{15}$ ,  $i_0 \sim i_{j-1}$  が、それぞれ、出力ライン  $o_8 \sim o_{15}$  に接続され、入力ライン  $i_j \sim i_{7+j}$  が、それぞれ、出力ライン  $o_0 \sim o_7$  に接続される。これにより、入力ライン  $i_8 \sim i_{7+j}$  に入力される  $j$  ビットの入力変数  $X_k$  が、後段の論理関数メモリ 4-k に入力されるとともに、入力ライン  $i_j \sim i_7$  に入力される  $8-j$  ビットの中間変数  $Y_k$  が後段の論理関数メモリ 4-k に入力される。従って、後段の論理関数メモリ 4-k に入力する入力変数  $X_k$  の変数



の個数 $|X_k|=n_k$ と、後段の論理関数メモリ 4-k に入力する中間変数 $Y_k$ の変数の個数 $|Y_k|=u_k$ とを、 $s=n_k+u_k=8$ の条件下で自由に変更することが可能となっている。

#### 【0064】

図5は図1の接続メモリ6の構成を表すブロック図である。

接続メモリ6-i ( $i \in \{1, 2, 3\}$ ) の内部には、 $m$ 個のメモリアレイ $CM_0 \sim CM_{m-1}$ が備えられている。各メモリアレイ $CM_0 \sim CM_{m-1}$ には、接続回路5-iの各シフト回路20-( $N_c-1$ )  $\sim$  20-0 (ここで、 $N_c$ は接続回路におけるシフト回路の段数を表し、図4では $N_c=4$ である。)のオン/オフ情報を表す接続情報( $s_{N_c-1}(j), \dots, s_0(j)$ ) ( $j \in \{0, \dots, m-1\}$ ) が記憶されている。尚、ここで、添字「(j)」 ( $j \in \{0, \dots, m-1\}$ ) は、j番目の目的論理関数 $f(j)$ に対応した接続情報であることを表す。

#### 【0065】

これらのメモリアレイ $CM_0 \sim CM_{m-1}$ の $N_c$ ビットの出力ライン $s_0 \sim s_{N_c-1}$ は、接続回路5-iの各シフト回路20-0  $\sim$  20-( $N_c-1$ )の制御ラインとなっている。

#### 【0066】

また、各接続メモリ6の内部には、ページデコーダ21が設けられている。ページデコーダ21は、演算制御部10から入力されるページ選択番号 $p_k$ に従って、k番目のメモリアレイ $CM_k$ を選択する。ページデコーダ21により選択されたメモリアレイ $CM_k$ は、 $N_c$ ビットの接続情報( $s_{N_c-1}(k), \dots, s_0(k)$ )からなる接続制御信号を接続回路5に出力する。

#### 【0067】

以上のように構成された本実施形態に係るルックアップテーブル・カスケード論理回路において、以下その動作について説明する。

#### 【0068】

図8は実施形態1に係るルックアップテーブル・カスケード論理回路の動作を表すフローチャートである。

#### 【0069】

最初に、演算を行おうとする目的論理関数 $f(X)$ を関数分解した分解関数 $\{f_0, \dots$

$\{f_p\}$  ( $1 \leq p \leq 3$ ) を、各論理関数メモリ 4-0 ~ 4-p の第 k ページに書き込んでおく。尚、ここで、k はページ番号を示す。また、各入力選択メモリ 3-0 ~ 3-3 及び各接続メモリ 6-1 ~ 6-3 の第 k ページには、上記各分解関数  $\{f_0, \dots, f_p\}$  に対応した入力選択情報及び接続情報を書き込んでおく。尚、各メモリへの書き込み機能については、図 1 では図示していないが、通常のメモリの書き込み方法により行われる。以下では、 $p=3$  であるものとして説明する。

#### 【0070】

以上の書き込みがされた状態において、まず、演算制御部 10 は、各入力選択メモリ 3-0 ~ 3-3、及び接続メモリ 6-1 ~ 6-3 に対して、ページ選択番号  $p_k$  を設定する (S1)。

#### 【0071】

これにより、各入力選択メモリ 3-0 ~ 3-3 は、第 k ページに書き込まれた入力選択情報を入力変数選択回路 2-0 ~ 2-3 の制御ライン shf0 ~ shf3 に出力する (S2)。また、接続メモリ 6-1 ~ 6-3 は、第 k ページに書き込まれた接続情報  $(s_3^{(k)}, \dots, s_0^{(k)})$  を接続回路 5-1 ~ 5-3 の各制御ラインに出力する (S3)。

#### 【0072】

入力変数選択回路 2-0 ~ 2-3 は、制御ライン shf0 ~ shf3 に入力された入力選択情報に従って、入力端子 in(00) ~ in(16) の一部を出力端子 out(00) ~ out(07) と電氣的に接続する。接続回路 5-1 ~ 5-3 は、各制御ラインより入力される接続情報  $(s_3^{(k)}, \dots, s_0^{(k)})$  に従って、入力ライン i8 ~ i15 と出力ライン o0 ~ o7 とを接続する。

#### 【0073】

次に、演算制御部 10 は、入力変数レジスタ 1 により、入力変数 X の出力を行う (S4)。これにより、論理関数メモリ 4-0 及び接続回路 5-1 ~ 5-3 には、それぞれ、入力変数  $X_0, X_1, X_2, X_3$  が出力される。

#### 【0074】

これに伴い、まず論理関数メモリ 4-0 は、入力変数  $X_0$  に対する分解関数  $f_0$  の真理値  $f_0(X_0)$  を中間変数  $Y_1$  として出力する。この中間変数  $Y_1$  は、接続回路 5-1

を介して論理関数メモリ 4-1 の入力に伝達される。また、場合によっては、中間変数 $Y_1$ の一部は、外部出力ライン 7-1 へと伝達される。

#### 【0075】

論理関数メモリ 4-1 には、論理関数メモリ 4-0 から伝達された中間変数 $Y_1$ と、入力変数選択回路 2-1 及び接続回路 5-1 を介して入力変数レジスタ 1 から伝達される入力変数 $X_1$ が入力される。論理関数メモリ 4-1 は、入力変数 $X_1$ 及び中間変数 $Y_1$ に基づき、分解関数 $f_1$ の真理値 $f_1(X_1, Y_1)$ を中間変数 $Y_2$ として出力する。この中間変数 $Y_2$ は、接続回路 5-2 を介して論理関数メモリ 4-2 の入力に伝達される。また、場合によっては、中間変数 $Y_2$ の一部は、外部出力ライン 7-2 へと伝達される。

#### 【0076】

論理関数メモリ 4-2 には、論理関数メモリ 4-1 から伝達された中間変数 $Y_2$ と、入力変数選択回路 2-2 及び接続回路 5-2 を介して入力変数レジスタ 1 から伝達される入力変数 $X_2$ が入力される。論理関数メモリ 4-2 は、入力変数 $X_2$ 及び中間変数 $Y_2$ に基づき、分解関数 $f_2$ の真理値 $f_2(X_2, Y_2)$ を中間変数 $Y_3$ として出力する。この中間変数 $Y_3$ は、接続回路 5-3 を介して論理関数メモリ 4-3 の入力に伝達される。また、場合によっては、中間変数 $Y_3$ の一部は、外部出力ライン 7-3 へと伝達される。

#### 【0077】

論理関数メモリ 4-3 には、論理関数メモリ 4-2 から伝達された中間変数 $Y_3$ と、入力変数選択回路 2-3 及び接続回路 5-3 を介して入力変数レジスタ 1 から伝達される入力変数 $X_3$ が入力される。論理関数メモリ 4-3 は、入力変数 $X_3$ 及び中間変数 $Y_3$ に基づき、分解関数 $f_3$ の真理値 $f_3(X_3, Y_3)$ を出力変数 $f$ として出力する。この出力変数 $f$ は、外部出力ライン 7-4 に伝達される (S 5)。以上の処理は、前段から後段に向かって直列に伝達しながらパイプライン処理により行われる。そして、目的関数 $f$ の演算結果は、各外部出力ライン 7-1 ~ 7-4 より取り出される。

#### 【0078】

尚、本実施形態では接続回路 5-1 ~ 5-3 として、図 4 のシフト回路を使用

したが、接続回路 5-1～5-3 としては、図 7 に示したクロスバスイッチ 5-i' や図 8 に示したマルチプレクサ 5-i'' を使用することも可能である。特に、クロスバスイッチ 5-i' は、シフト回路に比べると切換制御ラインの数が多くなり回路が大きくなるという欠点はあるが、信号が通過するパストランジスタの段数を減らすことができるため、演算速度の高速化が可能となる。

#### 【0079】

また、本実施形態では、各論理関数メモリ 4-0～4-3 を非同期に動作させる例を示したが、本発明においては、各論理関数メモリ 4-0～4-3 をクロックで同期させて動作させてもよい。

#### 【0080】

次に、上記動作をより分かりやすく説明するため、具体的な例を用いてルックアップテーブル・カスケード論理回路の具体的な動作を説明する。

#### 【0081】

##### (例) 加算回路

ここでは、簡単な例として、二つの 8 ビットの二進数  $A=(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ ,  $B=(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$  の加算を行う加算器を、上記ルックアップテーブル・カスケード論理回路によって実行する例を説明する。尚、下位からの桁上げ（キャリー）入力ビットとして  $c_{in}$  も考慮して、以下のような演算を行う加算器を考える。

#### 【数 1】

$$\begin{array}{r}
 \begin{array}{cccccccc}
 a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\
 b_7 & b_6 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\
 + & & & & & & & c_{in}
 \end{array} \\
 \hline
 c_{out} \quad S_7 \quad S_6 \quad S_5 \quad S_4 \quad S_3 \quad S_2 \quad S_1 \quad S_0
 \end{array}$$

ここで、桁上げ入力ビット  $c_{in}$  とは、8 ビット以上の大きな数の加算を行う場合に、加算器を直列につなげて使用する際に、下位の加算器からの桁上げを表すビットである。桁上げ出力ビット  $c_{out}$  も、同様に、上位の加算器に出力する桁上げを表すビットである。

#### 【0082】

このような二つの 8 ビットの二進数 A, B の加算を行う論理関数を  $f=f(X)$  ( $X=(A, B)$ ) とすると、論理関数  $f$  は図 9 により表される。この関数は、図 10 のように、8 つの分解関数  $\{g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7\}$  に関数分解することが可能である。ここで、各分解関数は、(数 2) 又は (数 3) のような論理式で表される。

### 【数 2】

$$\begin{aligned} g_0 &= [c_{out}^{(0)}, S_0] \\ &= [a_0 b_0 \vee a_0 c_{in} \vee b_0 c_{in}, a_0 \oplus b_0 \oplus c_{in}] \end{aligned}$$

### 【数 3】

$$\begin{aligned} g_i &= [c_{out}^{(i)}, S_i] \\ &= [a_i b_i \vee a_i c_{out}^{(i-1)} \vee b_i c_{out}^{(i-1)}, a_i \oplus b_i \oplus c_{out}^{(i-1)}] \\ &\quad (i = 1, 2, 3, 4, 5, 6, 7) \end{aligned}$$

すなわち、各分解関数は、2 つの入力変数  $a_i, b_i$  及び桁上げを表す中間変数  $c_{out}^{(i-1)}$  の 2 を法とする和  $S_i$  及び桁上げの中間変数  $c_{out}^{(i)}$  を出力する論理関数となっている。

### 【0083】

これを、図 1 のような 4 つの論理関数メモリ 4-0 ~ 4-3 からなるルックアップテーブル・カスケード論理回路で実現する場合には、図 11 に示したように、分解関数  $\{g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7\}$  を、4 つの分解関数組  $\{g_0, g_1\}$ ,  $\{g_2, g_3\}$ ,  $\{g_4, g_5\}$ ,  $\{g_6, g_7\}$  に分ける。そして、図 12 に示したように、各々の分解関数組を 1 つの分解関数  $f_0, f_1, f_2, f_3$  で表す。ここで、各分解関数は、(数 4) 又は (数 5) のような論理式で表される 3 変数入力 3 変数出力論理関数である。



## 【数 4】

$$\begin{aligned}
 f_0 &= [c_{out}^{(0)}, S_1, S_0] \\
 &= [a_1 b_1 \vee (a_1 \vee b_1)(a_0 b_0 \vee a_0 c_{in} \vee b_0 c_{in}), \\
 &\quad a_1 \oplus b_1 \oplus (a_0 b_0 \vee a_0 c_{in} \vee b_0 c_{in}), a_0 \oplus b_0 \oplus c_{in}]
 \end{aligned}$$

## 【数 5】

$$\begin{aligned}
 f_i &= [c_{out}^{(i)}, S_{2i+1}, S_{2i}] \\
 &= [a_{2i+1} b_{2i+1} \vee (a_{2i+1} \vee b_{2i+1})(a_{2i} b_{2i} \vee a_{2i} c_{out}^{(i-1)} \vee b_{2i} c_{out}^{(i-1)}), \\
 &\quad a_{2i+1} \oplus b_{2i+1} \oplus (a_{2i} b_{2i} \vee a_{2i} c_{out}^{(i-1)} \vee b_{2i} c_{out}^{(i-1)}), a_{2i} \oplus b_{2i} \oplus c_{out}^{(i-1)}] \\
 &\quad (i = 1, 2, 3)
 \end{aligned}$$

## 【0 0 8 4】

各々の分解関数 $f_i$ の真理値表は、図 1 3 (a) のようになる。そこで、図 1 3 (b) の真理値表を、論理関数メモリ 4-0 の第 0 ページに記憶させ、図 1 3 (c) の真理値表 (ルックアップテーブル) を論理関数メモリ 4-1 ~ 4-3 の第 0 ページに記憶させる。ここで、それぞれ、 $S_{2i}$ ,  $S_{2i+1}$ ,  $c_{out}^{(i)}$  ( $i=1, 2, 3$ ) は、それぞれ接続回路 5 - (i+1) の入力ライン $i_2, i_3, i_4$ に出力されるように論理関数メモリの出力ビットを割り当てる。

## 【0 0 8 5】

各入力選択メモリ 3-0 ~ 3-3 の第 0 ページには、それぞれ、0 ビットシフト、5 ビットシフト、9 ビットシフト、13 ビットシフトの情報を記憶させておく。すなわち、各入力選択メモリ 3-0 ~ 3-3 の入力選択情報 (shf0, shf1, shf2, shf3) として、それぞれ、(0, 0, 0, 0)、(0, 1, 0, 1)、(1, 0, 0, 1)、(1, 1, 0, 1) を記憶させておく。

## 【0 0 8 6】

各接続メモリ 6-1 ~ 6-3 の第 0 ページには、接続情報として 5 ビット分シフトさせる情報を記憶させる。すなわち、接続情報として $(s_3, s_2, s_1, s_0) = (0, 1, 0, 1)$ を記憶させておく。

## 【0 0 8 7】

以上のようにプログラムされた状態において、以下、ルックアップテーブル。

カスケード論理回路の演算処理を説明する。

【 0 0 8 8 】

まず、入力変数レジスタ 1 には、変数  $A, B, c_{in}$  が入力され記憶される。そして、演算制御部 1 0 は、入力選択メモリ 3-0 ~ 3-3、論理関数メモリ 4-0 ~ 4-3、及び接続メモリ 6-1 ~ 6-3 に対してページ選択番号  $p_k$  として第 0 ページを設定する。

【 0 0 8 9 】

これにより、各入力選択メモリ 3-0 ~ 3-3 は、第 0 ページに書き込まれた入力選択情報  $(shf0, shf1, shf2, shf3) = (0, 0, 0, 0), (0, 1, 0, 1), (1, 0, 0, 1), (1, 1, 0, 1)$  を、入力変数選択回路 2-0 ~ 2-3 の制御ライン  $shf0 \sim shf3$  に出力する。入力変数選択回路 2-0 ~ 2-3 は、制御ライン  $shf0 \sim shf3$  に入力された入力選択情報に従って、入力端子  $in(00) \sim in(16)$  の一部を出力端子  $out(00) \sim out(07)$  と電氣的に接続する。

【 0 0 9 0 】

具体的には、入力選択メモリ 3-0 は、 $(shf0, shf1, shf2, shf3) = (0, 0, 0, 0)$  なので、 $in(00) \sim in(07)$  を  $out(00) \sim out(07)$  に接続する。入力選択メモリ 3-1 は、 $(shf0, shf1, shf2, shf3) = (0, 1, 0, 1)$  なので、5 ビットシフトさせて、 $in(05) \sim in(12)$  を  $out(00) \sim out(07)$  に接続する。入力選択メモリ 3-2 は、 $(shf0, shf1, shf2, shf3) = (1, 0, 0, 1)$  なので、9 ビットシフトさせて、 $in(09) \sim in(16)$  を  $out(00) \sim out(07)$  に接続する。入力選択メモリ 3-3 は、 $(shf0, shf1, shf2, shf3) = (1, 1, 0, 1)$  なので、13 ビットシフトさせて、 $in(13) \sim in(16)$  を  $out(00) \sim out(03)$  に接続する。このとき、 $out(04) \sim out(07)$  には入力変数は伝達されない。

【 0 0 9 1 】

また、接続メモリ 6-1 ~ 6-3 は、各々、第 0 ページに書き込まれた接続情報  $(s_3^{(0)}, \dots, s_0^{(0)}) = (0, 1, 0, 1)$  を接続回路 5-1 ~ 5-3 の各制御ラインに出力する。接続回路 5-1 ~ 5-3 は、各制御ラインより入力される接続情報  $(s_3^{(k)}, \dots, s_0^{(k)})$  に従って、入力ライン  $i_8 \sim i_{15}$  と出力ライン  $o_0 \sim o_7$  とを接続する。具体的には、各接続メモリ 6-1 ~ 6-3 は、4 ビットシフトさせて、 $i_4 \sim i_{11}$  をそれぞれ

れ $i_0 \sim i_7$ に接続し、 $i_{12} \sim i_{15}$ 、 $i_0 \sim i_3$ をそれぞれ $o_8 \sim o_{15}$ に接続する。

### 【 0 0 9 2 】

次に、演算制御部 1 0 は、入力変数レジスタ 1 により、入力変数 $X$ の出力を行う。このとき、入力変数選択回路 2 - 1  $\sim$  2 - 3 の入力端子には、それぞれ、（表 1）のように入力変数の値が入力される。

【表 1】

$i(00)$	$i(01)$	$i(02)$	$i(03)$	$i(04)$	$i(05)$	$i(06)$	$i(07)$	$i(08)$
$c_{in}$	$a_1$	$a_0$	$b_1$	$b_0$	$a_3$	$a_2$	$b_3$	$b_2$
$i(09)$	$i(10)$	$i(11)$	$i(12)$	$i(13)$	$i(14)$	$i(15)$	$i(16)$	
$a_5$	$a_4$	$b_5$	$b_4$	$a_7$	$a_6$	$b_7$	$b_6$	

### 【 0 0 9 3 】

入力変数選択回路 2 - 0 は、 $(out(00), out(01), out(02), out(03), out(04)) = (c_{in}, a_1, a_0, b_1, b_0)$ を出力し、これを論理関数メモリ 4 - 0 に入力する。ここで、出力 $out(05) \sim out(07)$ は使用されないので省略した。

### 【 0 0 9 4 】

論理関数メモリ 4 - 0 は、第 0 ページに記憶されたルックアップテーブル $LUT_0$ のアドレス $(i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7) = (c_{in}, a_1, a_0, b_1, b_0, -, -, -)$ に記憶された真理値 $f_0(c_{in}, a_1, a_0, b_1, b_0)$ の値 $(o_0, o_1, o_2, o_3, o_4, o_5, o_6, o_7) = (-, -, S_0, S_1, c_{out}(0), -, -, -)$ を接続回路 5 - 1 に出力する。ここで、「-」はドントケア（0でも1でもよい。）を表す。このとき、接続回路 5 - 1 の各入力ライン $(i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7)$ には $(-, -, S_0, S_1, c_{out}(0), -, -, -)$ が入力される。

### 【 0 0 9 5 】

一方、入力変数選択回路 2 - 1 は、 $(out(00), out(01), out(02), out(03)) = (a_3, a_2, b_3, b_2)$ を出力し、これを接続回路 5 - 1 の入力ライン $(i_8, i_9, i_{10}, i_{11})$ に入力する。

### 【 0 0 9 6 】

接続回路 5 - 1 からは、それぞれ、（表 2）の値が出力される。

【表 2】

$o_0$	$o_1$	$o_2$	$o_3$	$o_4$	$o_5$	$o_6$	$o_7$
$c_{out}^{(0)}$	-	-	-	$a_3$	$a_2$	$b_3$	$b_2$
$o_8$	$o_9$	$o_{10}$	$o_{11}$	$o_{12}$	$o_{13}$	$o_{14}$	$o_{15}$
-	-	-	-	-	-	$S_0$	$S_1$

## 【0097】

接続回路 5-1 の出力ライン  $o_8 \sim o_{15}$  は、外部出力ライン 7-1 として取り出される。従って、演算結果  $S_0, S_1$  が外部出力ライン 7-1 のうちの 2 本に出力される。

## 【0098】

次に、論理関数メモリ 4-1 は、第 0 ページに記憶されたルックアップテーブル  $LUT_0$  のアドレス  $(i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7) = (c_{out}^{(0)}, -, -, -, a_3, a_2, b_3, b_2)$  に記憶された真理値  $f_1(c_{out}^{(0)}, a_3, a_2, b_3, b_2)$  の値  $(o_0, o_1, o_2, o_3, o_4, o_5, o_6, o_7) = (-, -, S_2, S_3, c_{out}^{(1)}, -, -, -)$  を接続回路 5-2 に出力する。このとき、接続回路 5-2 の各入力ライン  $(i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7)$  には  $(-, -, S_2, S_3, c_{out}^{(1)}, -, -, -)$  が入力される。

## 【0099】

一方、入力変数選択回路 2-2 は、 $(out(00), out(01), out(02), out(03)) = (a_5, a_4, b_5, b_4)$  を出力し、これを接続回路 5-2 の入力ライン  $(i_8, i_9, i_{10}, i_{11})$  に入力する。

## 【0100】

接続回路 5-2 からは、それぞれ、(表 3) の値が出力される。

【表 3】

$o_0$	$o_1$	$o_2$	$o_3$	$o_4$	$o_5$	$o_6$	$o_7$
$c_{out}^{(1)}$	-	-	-	$a_5$	$a_4$	$b_5$	$b_4$
$o_8$	$o_9$	$o_{10}$	$o_{11}$	$o_{12}$	$o_{13}$	$o_{14}$	$o_{15}$
-	-	-	-	-	-	$S_2$	$S_3$

## 【0 1 0 1】

接続回路 5-2 の出力ライン  $o_8 \sim o_{15}$  は、外部出力ライン 7-2 として取り出される。従って、演算結果  $S_2, S_3$  が外部出力ライン 7-2 のうちの 2 本に出力される。

## 【0 1 0 2】

次に、論理関数メモリ 4-2 は、第 0 ページに記憶されたルックアップテーブル  $LUT_0$  のアドレス  $(i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7) = (c_{out}^{(1)}, -, -, -, a_5, a_4, b_5, b_4)$  に記憶された真理値  $f_2(c_{out}^{(1)}, a_5, a_4, b_5, b_4)$  の値  $(o_0, o_1, o_2, o_3, o_4, o_5, o_6, o_7) = (-, -, S_4, S_5, c_{out}^{(2)}, -, -, -)$  を接続回路 5-3 に出力する。このとき、接続回路 5-3 の各入力ライン  $(i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7)$  には  $(-, -, S_4, S_5, c_{out}^{(2)}, -, -, -)$  が入力される。

## 【0 1 0 3】

一方、入力変数選択回路 2-3 は、 $(out(00), out(01), out(02), out(03)) = (a_7, a_6, b_7, b_6)$  を出力し、これを接続回路 5-3 の入力ライン  $(i_8, i_9, i_{10}, i_{11})$  に入力する。

## 【0 1 0 4】

接続回路 5-3 からは、それぞれ、(表 4) の値が出力される。

【表 4】

$o_0$	$o_1$	$o_2$	$o_3$	$o_4$	$o_5$	$o_6$	$o_7$
$c_{out}^{(2)}$	-	-	-	$a_7$	$a_6$	$b_7$	$b_6$
$o_8$	$o_9$	$o_{10}$	$o_{11}$	$o_{12}$	$o_{13}$	$o_{14}$	$o_{15}$
-	-	-	-	-	-	$S_4$	$S_5$

## 【0 1 0 5】

接続回路 5-3 の出力ライン  $o_8 \sim o_{15}$  は、外部出力ライン 7-3 として取り出される。従って、演算結果  $S_4, S_5$  が外部出力ライン 7-3 のうちの 2 本に出力される。

## 【0 1 0 6】

最後に、論理関数メモリ 4-3 は、第 0 ページに記憶されたルックアップテー



ブルLUT<sub>0</sub>のアドレス( $i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7$ ) = ( $c_{out}^{(2)}, -, -, -, a_7, a_6, b_7, b_6$ ) に記憶された真理値 $f_3(c_{out}^{(2)}, a_7, a_6, b_7, b_6)$ の値( $o_0, o_1, o_2, o_3, o_4, o_5, o_6, o_7$ ) = ( $-, -, S_6, S_7, c_{out}^{(3)}, -, -, -$ )を出力ライン7-4に出力する。これにより、総ての演算結果が出力ライン7-1~7-4に出力されて演算が終了する。

### 【0107】

(実施形態2)

図14は本発明の実施形態2に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

### 【0108】

本実施形態においては、入力変数選択回路2-1~2-3の出力ラインの一部を、接続回路5-1~5-3を介することなく、論理関数メモリ4-1~4-3の入力に直接接続したことを特徴とする。論理関数メモリ4-1~4-3の入力には、最低でも1つの入力変数が入力されることから、入力変数選択回路2-1~2-3の出力ラインの最低1本は、必ず論理関数メモリ4-1~4-3の入力に接続される。従って、最初から入力変数選択回路2-1~2-3の出力ラインの一部を、接続回路5-1~5-3を介さずに論理関数メモリ4-1~4-3の入力に直接接続しておけば、接続回路5-1~5-3の入力ラインの本数を減らすことができる。その結果、接続回路5-1~5-3を小型化することが可能となる。また、接続回路5-1~5-3を図4のようなシフト回路により構成する場合には、シフトの段数を減らすことが可能であり、演算速度の高速化及び低消費電力化を図ることができる。

### 【0109】

(実施形態3)

図15は本発明の実施形態3に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

### 【0110】

本実施の形態においては、接続回路5-1~5-5に入力する入力変数を選択する入力変数選択回路2-1~2-3に加えて、接続回路5-1~5-5を介さずに、論理関数メモリ4-1~4-3に直接入力する入力変数の選択を行う入力変数選択

回路 30-1 ~ 30-3 及びその入力選択情報を記憶する入力選択メモリ 31-1 ~ 31-3 を設けたことを第 1 の特徴とする。また、前段の論理関数メモリ 4- (i-1) ( $i \in \{1, 2, 3\}$ ) から後段の論理関数メモリ 4-i に入力される中間変数の一部を、接続回路 5-i を介さずに直接入力するように構成したことを第 2 の特徴とする。

#### 【0111】

このように構成することで、接続回路 5-1 ~ 5-3 の入力ラインの本数を減らすことができる。その結果、接続回路 5-1 ~ 5-3 を小型化することが可能となる。また、接続回路 5-1 ~ 5-3 を図 4 のようなシフタ回路により構成する場合には、シフタの段数を減らすことが可能であり、演算速度の高速化を図ることができる。

#### 【0112】

##### (実施形態 4)

図 16 は本発明の実施形態 4 に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図、図 17 は図 16 の出力レジスタ及び出力デコーダの構成を表すブロック図、図 18 は図 17 の記憶素子の構成を表すブロック図である。本実施形態においては、外部出力ライン 7-1 ~ 7-3 に出力された出力変数を一時的に記憶する出力変数レジスタ 51、及び出力変数レジスタ 51 が出力変数を取り込むためのロード信号を出力する出力選択デコーダ 52 を備えたことを特徴とする。他の構成については、図 1 と同様であるため、説明は省略する。尚、本実施形態においては各論理関数メモリ 4-0 ~ 4-3 はクロックにより同期して動作するものとする。

#### 【0113】

出力変数レジスタ 51 は、32 個の記憶素子  $M(i, j)$  ( $i=1, 2, 3, 4, j=0, 1, 2, 3, 4, 5, 6, 7$ ) を備えている。各記憶素子  $M(i, j)$  は、データを保持する D フリップ・フロップ 53 と 2 入力 1 出力のマルチプレクサ 54 とから構成されている。D フリップ・フロップ 53 には、共通のクロック信号 Clock が入力される。このクロック信号 Clock が 1 のときに、D フリップ・フロップ 53 はデータ入力 D に入力された値をラッチする。マルチプレクサ 54 の出力は D フリップ・フ

ロップ 4 3 のデータ入力 D に接続されている。また、D フリップ・フロップ 5 3 の出力 Q は、マルチプレクサ 5 4 の 0 側の入力 D 0 に接続されている。マルチプレクサ 5 4 の 1 側の入力 D 1 は、外部出力ライン 7-i の j 番目のラインに接続されている。マルチプレクサ 5 4 は、ロード信号 Load により制御され、Load = 0 のときに 0 側の入力を選択し、Load = 1 のときに 1 側の入力を選択する。

#### 【0114】

出力選択デコーダ 5 2 には、2 ビットの出力選択信号 t が演算制御部 1 0 から入力され、4 つの出力選択信号  $T_1 \sim T_4$  を出力する。t = (0, 0) のときは、( $T_1, T_2, T_3, T_4$ ) = (1, 0, 0, 0)、t = (0, 1) のときは、( $T_1, T_2, T_3, T_4$ ) = (0, 1, 0, 0)、t = (1, 0) のときは、( $T_1, T_2, T_3, T_4$ ) = (0, 0, 1, 0)、t = (1, 1) のときは、( $T_1, T_2, T_3, T_4$ ) = (0, 0, 0, 1) が出力される。各出力選択信号  $T_i$  ( $i = 1, 2, 3, 4$ ) は、各記憶素子 M (i, j) にロード信号 Load として入力される。

#### 【0115】

演算制御部 1 0 は、演算開始時には、t = (0 0) として、論理関数メモリ 4-0 が演算結果を出力した後に t = (0 1)、論理関数メモリ 4-1 が演算結果を出力した後に t = (1 0)、論理関数メモリ 4-2 が演算結果を出力した後に t = (1 1) のように出力選択信号 t を切り換えて、演算結果を出力変数レジスタ 5 1 にラッチする。そして、演算が終了した時点で、出力変数レジスタ 5 1 に記憶された出力変数を読み出すことで、演算結果を得ることができる。

#### 【0116】

(実施形態 5)

図 1 9 は本発明の実施形態 5 に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

#### 【0117】

本実施形態においては、第 1 の出力選択手段 2 5、第 2 の出力選択手段 2 6、及び入力変数選択回路 2-4 を備えていることを特徴とする。尚、他の構成については、図 1 と同様であるため、説明は省略する。

#### 【0118】

第2の出力選択手段26は、入力変数選択回路2-4から入力される入力変数の値に従って、最終段の論理関数メモリ4-3から出力される出力変数のうちの全部又は一部を選択して出力する。尚、入力変数選択回路2-4は、図2に示したものと同様のシフト回路で構成されている。但し、本実施形態では、最終段の論理関数メモリ4-3の出力が8ビットであるため、入力変数選択回路2-4に使用されるシフト回路の出力は、3ビットとされる。一般には、入力変数選択回路2-4の出力は、最終段の論理関数メモリ4-3の出力の数Nに対して、

【数6】

$$\lceil \log_2 N \rceil$$

とされる。以下、入力変数選択回路2-4の出力out(00)~out(02) (図2参照)を、それぞれ、 $F_0$ ,  $F_1$ ,  $F_2$ で表す。

【0119】

第1の出力選択手段25は、マルチプレクサにより構成されており、演算制御部10から出力される出力選択信号tに従って、各論理関数メモリ4-0~4-2及び第2の出力選択手段26から出力される出力変数のうちの何れかを選択して出力する。

【0120】

ここで、出力選択信号tとは、第1の出力選択回路25が選択する出力ライン束の指定番号 ( $t \in \{(00), (01), (10), (11)\}$ ) を表す2ビットの信号である。第1の出力選択回路25は、 $t = (00)$ のときは接続回路5-1の出力ライン束 (図4のo8~o15) を選択し、 $t = (01)$ のときは接続回路5-2の出力ライン束を選択し、 $t = (10)$ のときは接続回路5-3の出力ライン束を選択し、 $t = (11)$ のときは第2の出力選択回路26の出力ライン束を選択する。

【0121】

図20は図19の第2の出力選択回路の構成を表すブロック図である。尚、この回路図は、動作原理を説明するために、簡略化して記載している。

【0122】

第2の出力選択回路26は、図20に示したように、2入力1出力のマルチプ

レクサ（以下、「MUX」という。）3 1～3 7を多段にカスケード状に接続し、各段において出力を、MUX 3 8～MUX 4 4を通して取り出すことができるように構成されている。尚、図 2 0では、説明の都合上、出力ラインのビット数 $N_c$ が8ビットの例を示しているが、 $N_c$ は8ビットに限られるものではない。

### 【0 1 2 3】

図 2 0において、第 2 の出力選択回路 2 6の入力ライン $y_0 \sim y_7$ は、論理関数メモリ 4 - 3の出力に接続される。この入力ライン $y_0 \sim y_7$ より、出力変数 $Y$ の値が入力される。また、出力選択回路 2 6の出力ライン $f(0) \sim f(7)$ からは、入力ライン $y_0 \sim y_7$ のうち選択された出力変数 $Y$ の値( $y_0, \dots, y_{|Y|-1}$ )が出力される。

### 【0 1 2 4】

入力ライン $y_0, y_1$ 、入力ライン $y_2, y_3$ 、入力ライン $y_4, y_5$ 、及び入力ライン $y_6, y_7$ は、それぞれMUX 3 1、MUX 3 2、MUX 3 3、及びMUX 3 4の入力側に接続されている。MUX 3 1、3 2、及びMUX 3 3、3 4の出力は、それぞれMUX 3 5、及びMUX 3 6の入力側に接続されている。MUX 3 5、3 6の出力は、MUX 3 7の入力に接続されている。

### 【0 1 2 5】

一方、（入力ライン $y_0$ 、MUX 3 1の出力）、（入力ライン $y_2$ 、MUX 3 2の出力）、（入力ライン $y_4$ 、MUX 3 3の出力）、及び（入力ライン $y_6$ 、MUX 3 4の出力）は、それぞれMUX 3 8、MUX 3 9、MUX 4 0、及びMUX 4 1の入力側に接続されている。また、（入力ライン $y_1$ 、MUX 3 5の出力）、（入力ライン $y_3$ 、MUX 3 7の出力）、及び（入力ライン $y_5$ 、MUX 3 6の出力）は、それぞれMUX 4 2、MUX 4 3、及びMUX 4 4の入力側に接続されている。

### 【0 1 2 6】

MUX 3 1～3 4は、共通の入力変数 $F_0$ により切換制御がされる。すなわち、 $F_0$ が“0”のときは、MUX 3 1、3 2、3 3、3 4は、それぞれ、入力ライン $y_0, y_2, y_4, y_6$ を選択し、 $F_0$ が“1”のときは、MUX 3 1、3 2、3 3、3 4は、それぞれ、入力ライン $y_1, y_3, y_5, y_7$ を選択する。

### 【0 1 2 7】

MUX 3 5、3 6は、共通の入力変数 $F_1$ により切換制御がされる。すなわち、 $F_1$



が“0”のときは、MUX 3 5, 3 6は、それぞれ、MUX 3 1, 3 3を選択し、 $F_1$ が“1”のときは、MUX 3 5, 3 6は、それぞれ、MUX 3 2, 3 4を選択する。

### 【0 1 2 8】

また、MUX 3 7は、入力変数 $F_2$ により切換制御がされる。すなわち、 $F_2$ が“0”のときはMUX 3 7は、MUX 3 5を選択し、 $F_2$ が“1”のときは、MUX 3 7は、MUX 3 6を選択する。

### 【0 1 2 9】

MUX 3 1～3 4によって、8本の入力ライン $y_0 \sim y_7$ のうちの4本が選択される。MUX 3 1～3 6によって、8本の入力ライン $y_0 \sim y_7$ のうちの2本が選択される。更に、MUX 3 1～3 7によって、8本の入力ライン $y_0 \sim y_7$ のうちの1本が選択される。これは、目的論理関数 $f$ の出力変数の個数 $|f|$ が4個以下の場合、論理関数メモリ 4-3の出力する中間変数 $Y$ に対して、更に入力変数( $F_0, F_1, F_2$ )による論理演算を行うことが可能であることを意味する。従って、論理関数メモリ 4-0～4-3により行う演算における入力変数の数を減らすことができる。

### 【0 1 3 0】

すなわち、例えば、目的論理関数 $f(x_0, x_1, \dots, x_{n-2}, x_{n-1})$ を、

### 【数 7】

$$x_{n-1}f'(x_0, x_1, \dots, x_{n-2}) \vee \bar{x}_{n-1}f'(x_0, x_1, \dots, x_{n-2})$$

のようにシャノン展開して、関数 $f'(x_0, x_1, \dots, x_{n-2})$ を4つの分解関数 $f_0(X_0)$ ,  $f_1(X_1, Y_1)$ ,  $f_2(X_2, Y_2)$ ,  $f_3(X_3, Y_3)$  (但し、 $X_0 \cup X_1 \cup X_2 \cup X_3 = \{x_0, x_1, \dots, x_{n-2}\}$ ) に関数分解して、最終段の分解関数 $f_3(X_3, Y_3)$ の出力数を4以下となるようにすれば、論理関数メモリ 4-0～4-3に入力する入力変数の個数を1個減らすことができる。そして、入力変数 $x_{n-1}$ は、入力変数 $F_0$ として、第2の出力変数選択回路 2 6に入力するようにすればよい。

### 【0 1 3 1】

同様に、目的論理関数 $f(x_0, x_1, \dots, x_{n-2}, x_{n-1})$ を、

## 【数 8】

$$x_{n-1}x_{n-2}f''(x_0, x_1, \dots, x_{n-3}) \vee x_{n-1}\bar{x}_{n-2}f''(x_0, x_1, \dots, x_{n-3}) \\ \vee \bar{x}_{n-1}x_{n-2}f''(x_0, x_1, \dots, x_{n-3}) \vee \bar{x}_{n-1}\bar{x}_{n-2}f''(x_0, x_1, \dots, x_{n-3})$$

のようにシャノン展開すれば、論理関数メモリ 4-0～4-3 に入力する入力変数の個数を 2 個減らすことができ、

## 【数 9】

$$x_{n-1}x_{n-2}x_{n-3}f'''(x_0, x_1, \dots, x_{n-4}) \vee x_{n-1}x_{n-2}\bar{x}_{n-3}f'''(x_0, x_1, \dots, x_{n-4}) \\ \vee x_{n-1}\bar{x}_{n-2}x_{n-3}f'''(x_0, x_1, \dots, x_{n-4}) \vee x_{n-1}\bar{x}_{n-2}\bar{x}_{n-3}f'''(x_0, x_1, \dots, x_{n-4}) \\ \vee \bar{x}_{n-1}x_{n-2}x_{n-3}f'''(x_0, x_1, \dots, x_{n-4}) \vee \bar{x}_{n-1}x_{n-2}\bar{x}_{n-3}f'''(x_0, x_1, \dots, x_{n-4}) \\ \vee \bar{x}_{n-1}\bar{x}_{n-2}x_{n-3}f'''(x_0, x_1, \dots, x_{n-4}) \vee \bar{x}_{n-1}\bar{x}_{n-2}\bar{x}_{n-3}f'''(x_0, x_1, \dots, x_{n-4})$$

のようにシャノン展開すれば、論理関数メモリ 4-0～4-3 に入力する入力変数の個数を 3 個減らすことができる。

## 【0132】

尚、MUX 3 1～MUX 3 7により選択された 8 ビット、4 ビット、2 ビット、又は 1 ビットの出力は、共通の 8 本の出力ラインによって第 1 の出力選択手段 2 5 に出力する必要がある。そこで、図 2 0 では、MUX 3 8～4 4 によって、上記各本数の出力ラインが選択されたときの出力を、共通の出力ライン  $f(0) \sim f(7)$  を通して出力されるように構成されている。MUX 3 8～4 4 は、共通の選択切換変数 Select によって切換制御がされる。尚、選択切換変数 Select の設定値は、出力変数の個数  $|f(X)|$  によって、あらかじめ選択切換メモリ 4 5 に記憶される。そして、演算制御部 1 0 の制御によって、Select に設定される。

## 【0133】

MUX 3 8 は、Select が “0” のとき入力ライン  $y_0$  を選択し、Select が “1” のとき MUX 3 1 の出力を選択し、選択したラインの信号値を出力ライン  $f(0)$  に出力する。MUX 3 9 は、Select が “0” のとき入力ライン  $y_2$  を選択し、Select が “1” のとき MUX 3 2 の出力を選択し、選択したラインの信号値を出力ライン  $f(2)$  に出力する。MUX 4 0 は、Select が “0” のとき入力ライン  $y_4$  を選択し、Select が “1” のとき MUX 3 3 の出力を選択し、選択したラインの信号値を出力ライン  $f(4)$  に出力する。MUX 4 1 は、Select が “0” のとき入力ライン  $y_6$  を選択し、Select

が“1”のときMUX 3 4の出力を選択し、選択したラインの信号値を出力ラインf(6)に出力する。

#### 【0 1 3 4】

MUX 4 2は、Selectが“0”のとき入力ラインy<sub>1</sub>を選択し、Selectが“1”のときMUX 3 5の出力を選択し、選択したラインの信号値を出力ラインf(1)に出力する。MUX 4 3は、Selectが“0”のとき入力ラインy<sub>3</sub>を選択し、Selectが“1”のときMUX 3 7の出力を選択し、選択したラインの信号値を出力ラインf(3)に出力する。MUX 4 4は、Selectが“0”のとき入力ラインy<sub>5</sub>を選択し、Selectが“1”のときMUX 3 6の出力を選択し、選択したラインの信号値を出力ラインf(5)に出力する。また、出力ラインf(7)は、入力ラインy<sub>7</sub>に直結されている。

#### 【0 1 3 5】

目的論理関数fの演算結果f(X)のビット数（出力の本数）が8のときは、Selectを“0”とする。これにより、出力ラインf(0)～f(7)には、入力ラインy<sub>0</sub>～y<sub>7</sub>に入力される出力変数Y=(y<sub>0</sub>, ..., y<sub>7</sub>)の値が出力される。出力の本数が4のときは、Selectを“1”に設定し、入力変数F<sub>0</sub>に“0”又は“1”を入力する。これにより、入力ラインy<sub>0</sub>～y<sub>7</sub>のうちF<sub>0</sub>=0のときは(y<sub>0</sub>, y<sub>2</sub>, y<sub>4</sub>, y<sub>6</sub>)の4本が、F<sub>0</sub>=1のときは(y<sub>1</sub>, y<sub>3</sub>, y<sub>5</sub>, y<sub>7</sub>)の4本が、出力ラインf(0), f(2), f(4), f(6)に出力される。出力の本数が2のときは、Selectを“1”とし、入力変数F<sub>0</sub>, F<sub>1</sub>に“0”又は“1”を設定する。これにより、(F<sub>0</sub>, F<sub>1</sub>)=(0, 0)のときは、出力ラインf(1), f(5)には入力ラインy<sub>0</sub>～y<sub>7</sub>のうち2本に入力される出力変数Y=(y<sub>0</sub>, y<sub>4</sub>)の値が出力される。(F<sub>0</sub>, F<sub>1</sub>)=(0, 0), (0, 1), (1, 1)のときは出力には(y<sub>1</sub>, y<sub>5</sub>), (y<sub>2</sub>, y<sub>6</sub>), (y<sub>3</sub>, y<sub>7</sub>)の値がそれぞれ出力される。出力の本数が1のときは、Selectを“1”とし、入力変数F<sub>0</sub>, F<sub>1</sub>, F<sub>2</sub>に“0”又は“1”を設定する。これにより、出力ラインf(3)には、入力ラインy<sub>0</sub>～y<sub>7</sub>のうち1本にF<sub>0</sub>, F<sub>1</sub>, F<sub>2</sub>で指定した入力が入力される。

#### 【0 1 3 6】

このように第2の出力選択回路2 6を設けることにより、出力変数|f(X)|の個数が、最終段の論理関数メモリ4-3の全出力ラインの本数の1/2以下の場合には、第2の出力選択回路2 6を用いて更に論理演算を行うことが可能である。

これによって、論理関数メモリ 4-0～4-3 において演算を行う分解関数の入力変数の総数を 1 個以上減らすことが可能となる。従って、ルックアップテーブル・カスケード論理回路全体で許容される入力変数の数を増やすことが可能となる。

#### 【0137】

尚、図 20 では、出力選択回路 26 内のマルチプレクサは 2 入力 1 出力のものを使用しているが、一般に、 $w$  入力 1 出力 ( $w \geq 2$ ) のマルチプレクサを使用することが可能である。

#### 【0138】

(実施形態 6)

図 21 は本発明の実施形態 6 に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

#### 【0139】

本実施形態においては、第 2 の出力選択回路 26 を、第 1 の出力選択回路 25 の後段側に接続したことを特徴とする。尚、第 1 の出力選択回路 25 及び第 2 の出力選択回路 26 の接続順序以外は、実施形態 5 と同様に構成されているので、各部の説明については省略する。

#### 【0140】

接続回路 5-1～5-3 及び論理関数メモリ 4-3 から出力される変数組は、第 1 の選択回路 25 において何れか一つが選択される。第 1 の選択回路 25 において選択された変数組は、入力変数 ( $F_1, F_2, F_3$ ) 及び選択切換変数 Select の値に基づいて、第 2 の選択回路 26 によりその総て又は一部が選択され、出力変数として出力される。

#### 【0141】

ここで、出力変数の個数が 2 の冪乗である任意の  $n$  変数論理関数  $f(x_0, x_1, \dots, x_{n-1})$  は、以下の形で表現することが可能である。

## 【数 10】

$$\begin{aligned}
 & f(x_0, x_1, \dots, x_{n-3}, x_{n-2}, x_{n-1}) \\
 &= x_{n-2}x_{n-1}f_0(x_0, x_1, \dots, x_{n-3}) \vee x_{n-2}\bar{x}_{n-1}f_1(x_0, x_1, \dots, x_{n-3}) \\
 & \vee \bar{x}_{n-2}x_{n-1}f_2(x_0, x_1, \dots, x_{n-3}) \vee \bar{x}_{n-2}\bar{x}_{n-1}f_3(x_0, x_1, \dots, x_{n-3})
 \end{aligned}$$

## 【0142】

従って、上記分解関数 $f_0 \sim f_3$ を論理関数メモリ 4-0 ~ 4-3 により実現し、各分解関数 $f_0 \sim f_3$ と $x_{n-2}, x_{n-1}$ 等との積項を第2の出力選択回路 26 により演算することで、 $n$ 変数の論理関数の演算を行うことが可能となる。

## 【0143】

## 【発明の効果】

以上のように、本発明によれば、接続回路と接続メモリとを用いて、それぞれの目的論理関数に応じて、前段の論理関数メモリの出力及び入力変数の入力と後段の論理関数メモリの入力との接続関係を変更することができる構成とした。これにより、使用可能な分解関数のレイル数及び入力変数の数の組み合わせの自由度が増し、多様な目的論理関数の論理回路を1つのルックアップテーブル・カスケード論理回路により設計することが可能となる。レイル数及び入力変数の数の組み合わせを最適化できるため、論理関数メモリの入力線数を極力少なくすることができ、メモリの使用効率も向上する。その結果、回路のレイアウト効率が向上する。従って、回路の小型化・高集積化を図ることができる。

## 【0144】

また、各論理関数メモリについて、アクセス可能なページを選択的に切り換えるページ切換手段を設け、目的論理関数に応じて、分解関数のルックアップテーブルの書き込みページを割り当てれば、簡単な構成により、1つのルックアップテーブル・カスケード論理回路により設計することが可能となる。

## 【0145】

また、接続回路が、前段の論理関数メモリの出力を、論理関数の演算結果を出力する出力ラインに接続可能としたことで、目的論理関数の分解関数の段数が少ない場合には、カスケードの途中の論理関数メモリの出力を出力変数として取り



出し、必要なメモリ量を削減し、演算速度を高速化することが可能となる。

【0146】

更に、各接続回路及び各論理関数メモリに対して入力する入力変数を、各入力選択回路により個別に選択することができる。従って、各論理関数メモリの演算を同時に行うことが可能であり、パイプライン処理が可能となる。

【図面の簡単な説明】

【図1】 本発明の実施形態1に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図2】 図1の入力変数選択回路の構成を表す回路図である。

【図3】 図1の論理関数メモリの構成を表すブロック図である。

【図4】 図1の接続回路の周辺の回路ブロック図である。

【図5】 図1の接続メモリの構成を表すブロック図である。

【図6】 実施形態1に係るルックアップテーブル・カスケード論理回路の動作を表すフローチャートである。

【図7】 図1の接続回路にクロスバスイッチを使用した場合の回路ブロック図である。

【図8】 図1の接続回路にマルチプレクサを使用した場合の回路ブロック図である。

【図9】 8ビット加算器の論理関数を表す図である。

【図10】 8つの分解関数に関数分解した8ビット加算器を表す図である。

【図11】 8つの分解関数に関数分解した8ビット加算器を表す図である。

【図12】 4つの分解関数に関数分解した8ビット加算器を表す図である。

【図13】 4つの分解関数に関数分解した8ビット加算器の各分解関数の真理値である。

【図14】 本発明の実施形態2に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図 15】 本発明の実施形態 3 に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図 16】 本発明の実施形態 4 に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図 17】 図 16 の出力レジスタ及び出力デコーダの構成を表すブロック図である。

【図 18】 図 17 の記憶素子の構成を表すブロック図である。

【図 19】 本発明の実施形態 5 に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図 20】 図 19 の第 2 の出力選択回路の構成を表すブロック図である。

【図 21】 本発明の実施形態 6 に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図 22】 ルックアップテーブル・カスケード論理回路の原理を表す図である。

#### 【符号の説明】

- 1 入力変数レジスタ
- 2-0 ~ 2-3 入力変数選択回路
- 3-0 ~ 3-3 入力選択メモリ
- 4-0 ~ 4-3 論理関数メモリ
- 5-0 ~ 5-2 接続回路
- 6-0 ~ 6-2 接続メモリ
- 7-1 ~ 7-3 外部出力ライン
- 10 演算制御部
- 11-0 ~ 11-3 シフト回路
- 15 ページデコーダ
- 16 アドレスデコーダ
- 17 入力ライン
- 18 出力ライン
- 20-0 ~ 20-2 シフタ回路

2 1 ページデコーダ

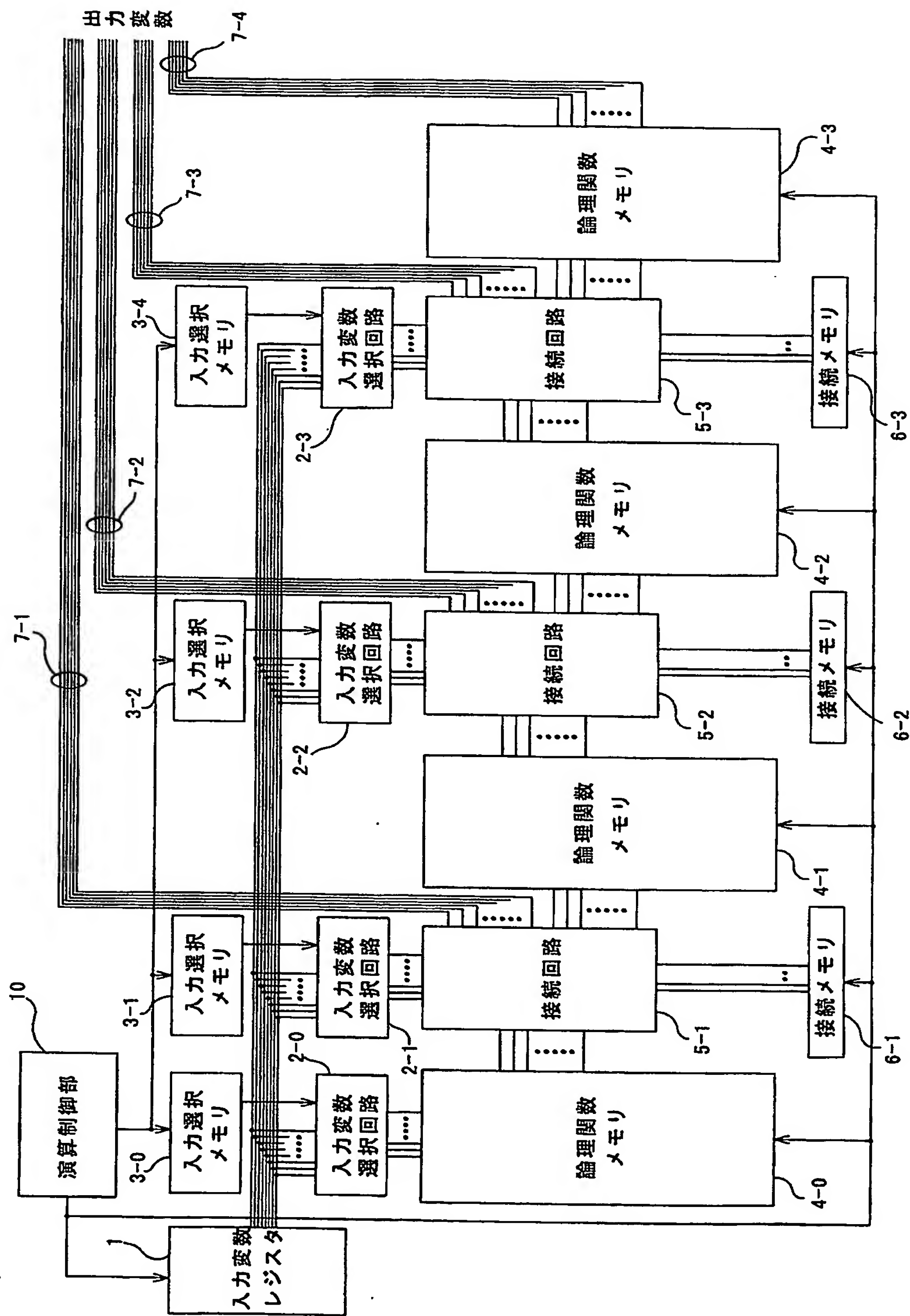
2 5 第 1 の出力選択回路

2 6 第 2 の出力選択回路

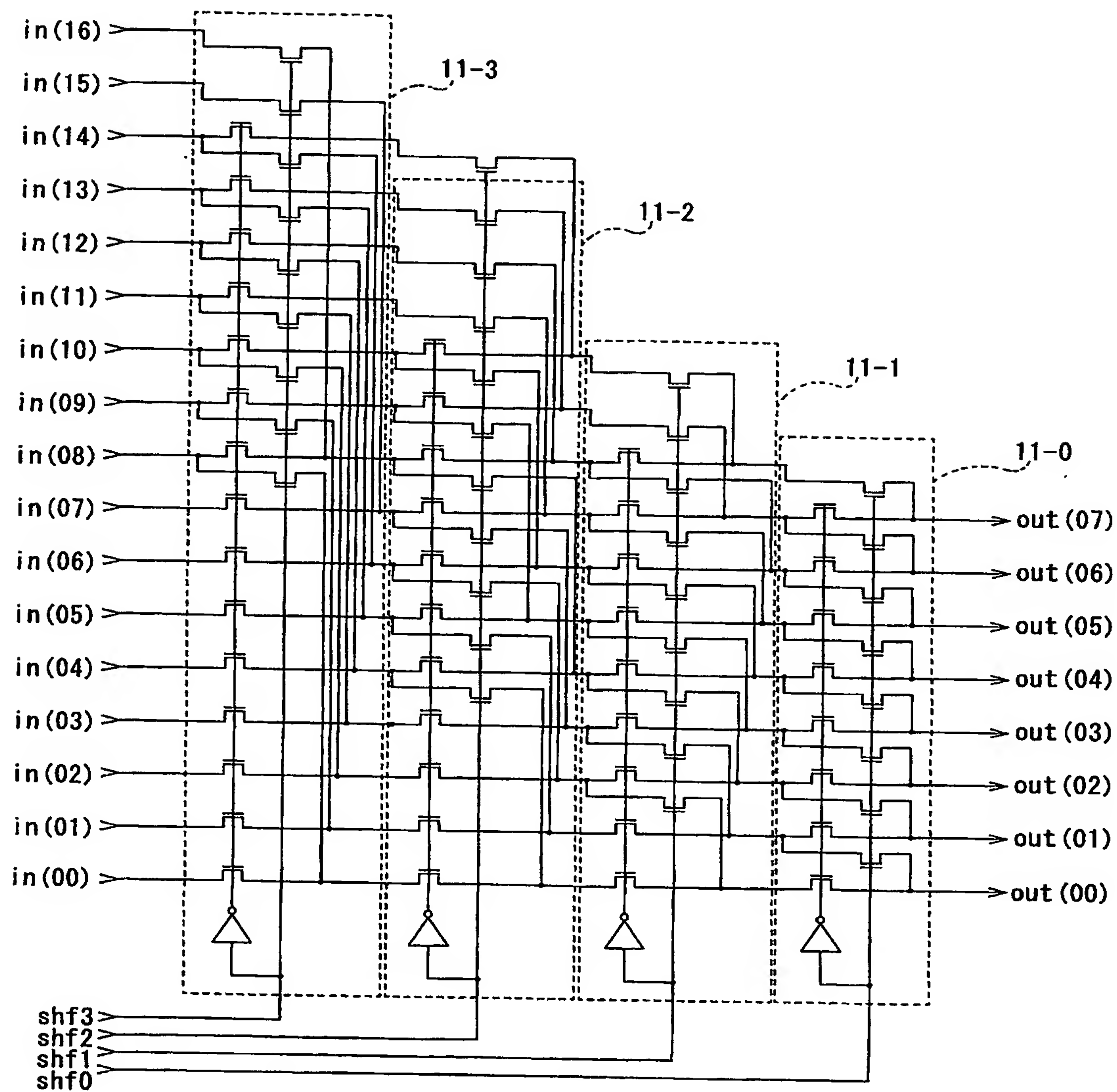
3 1 ~ 4 4 マルチプレクサ

【書類名】 図面

【図 1】

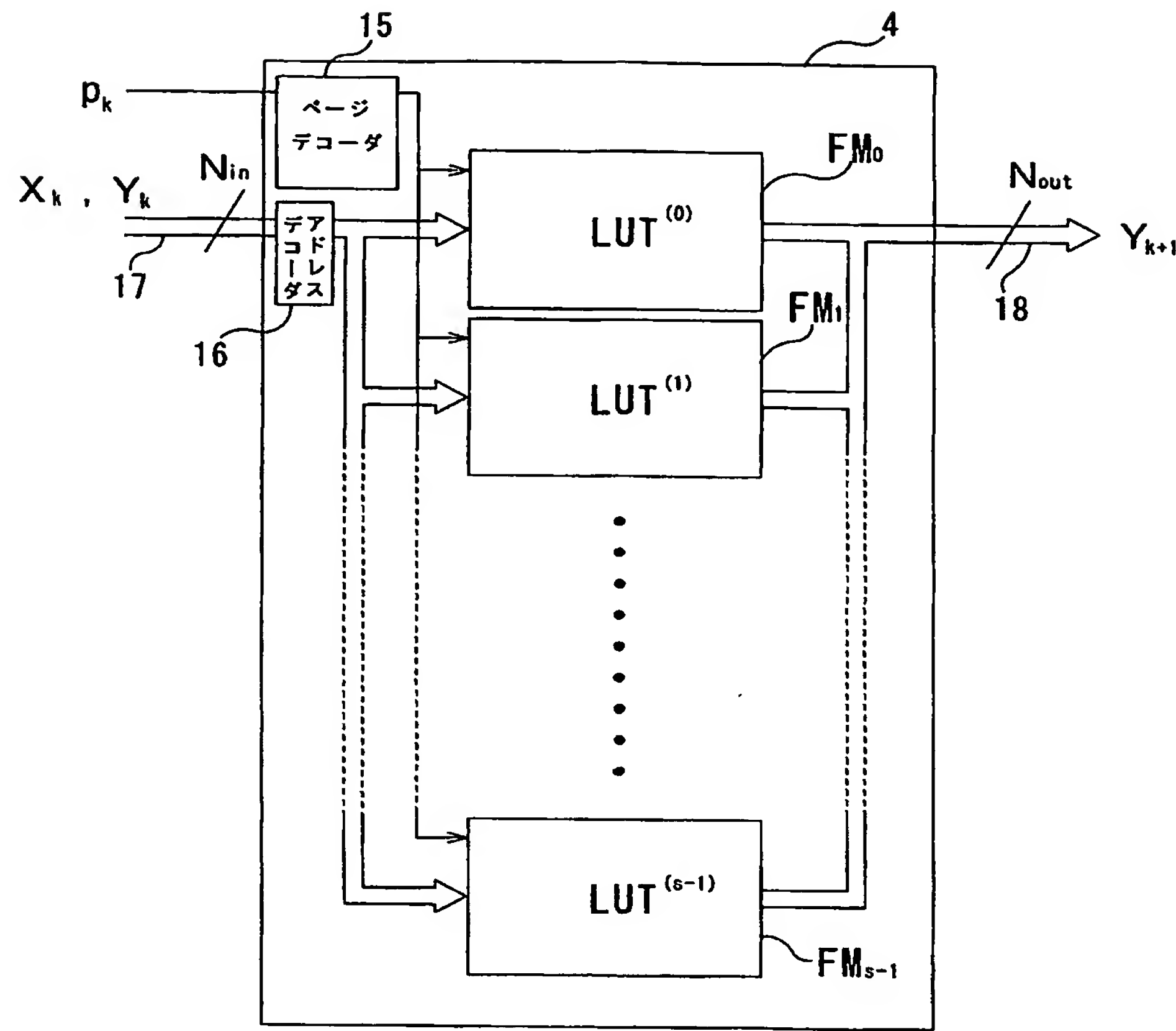


【図 2】

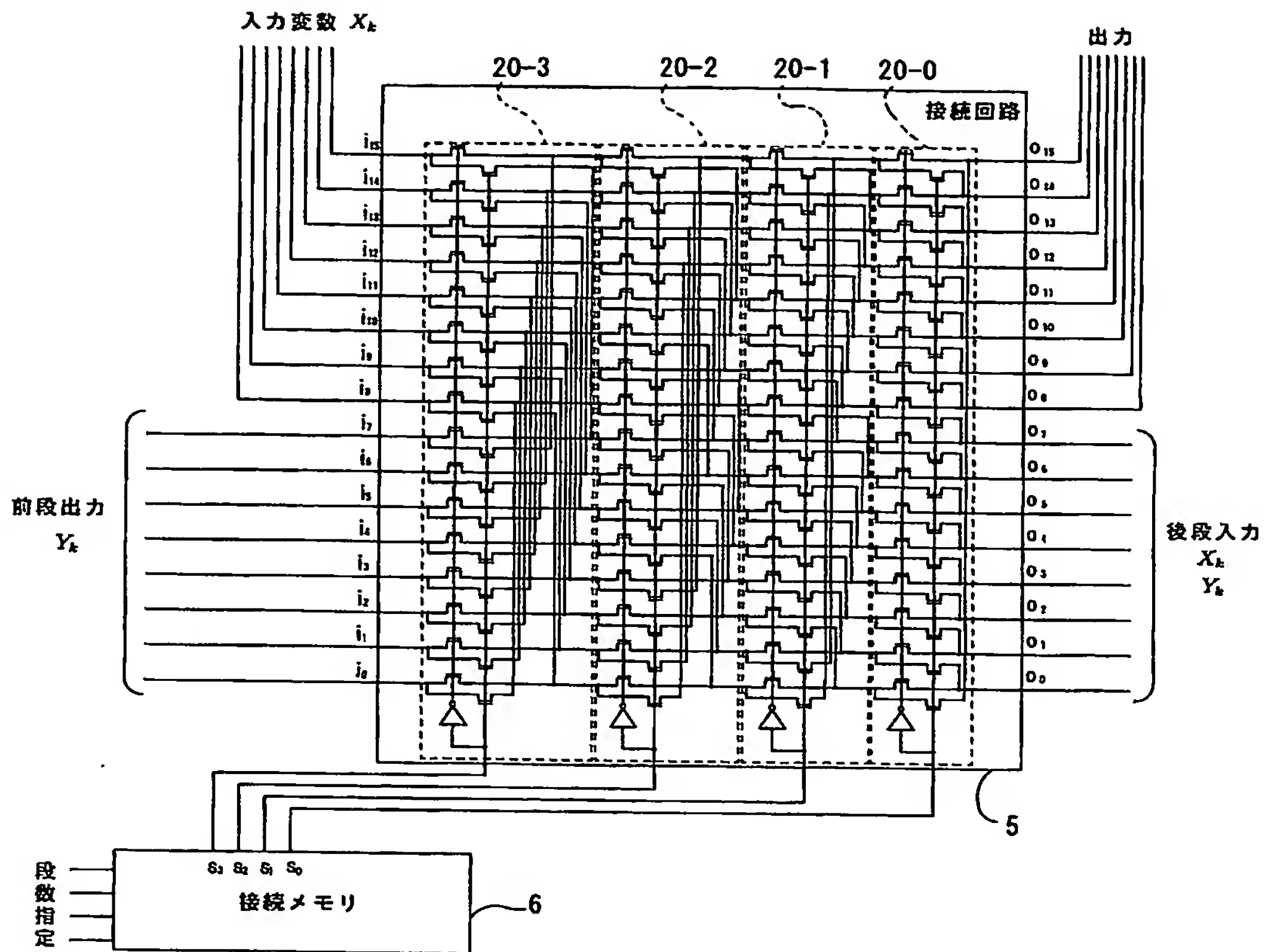




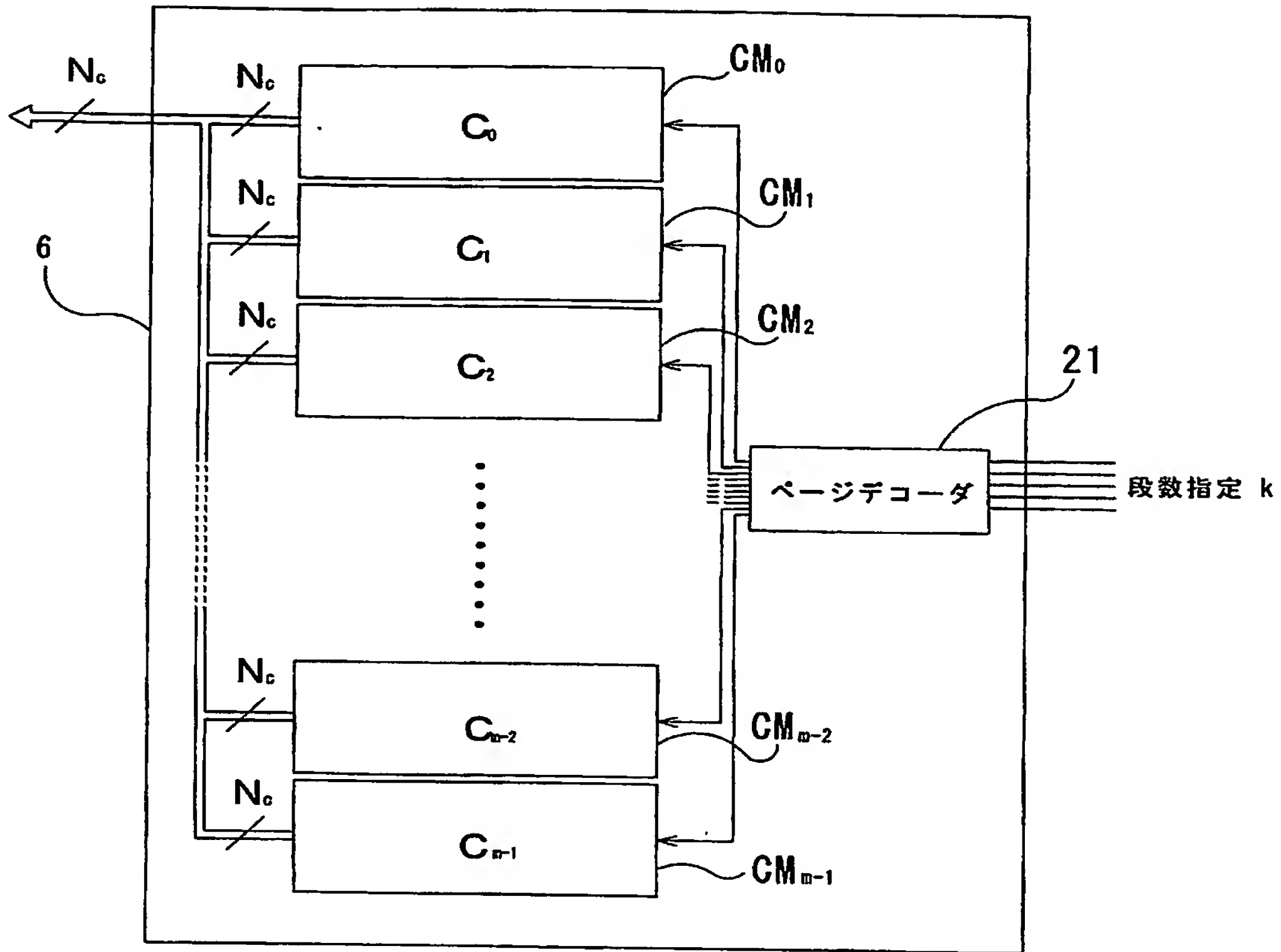
【図 3】



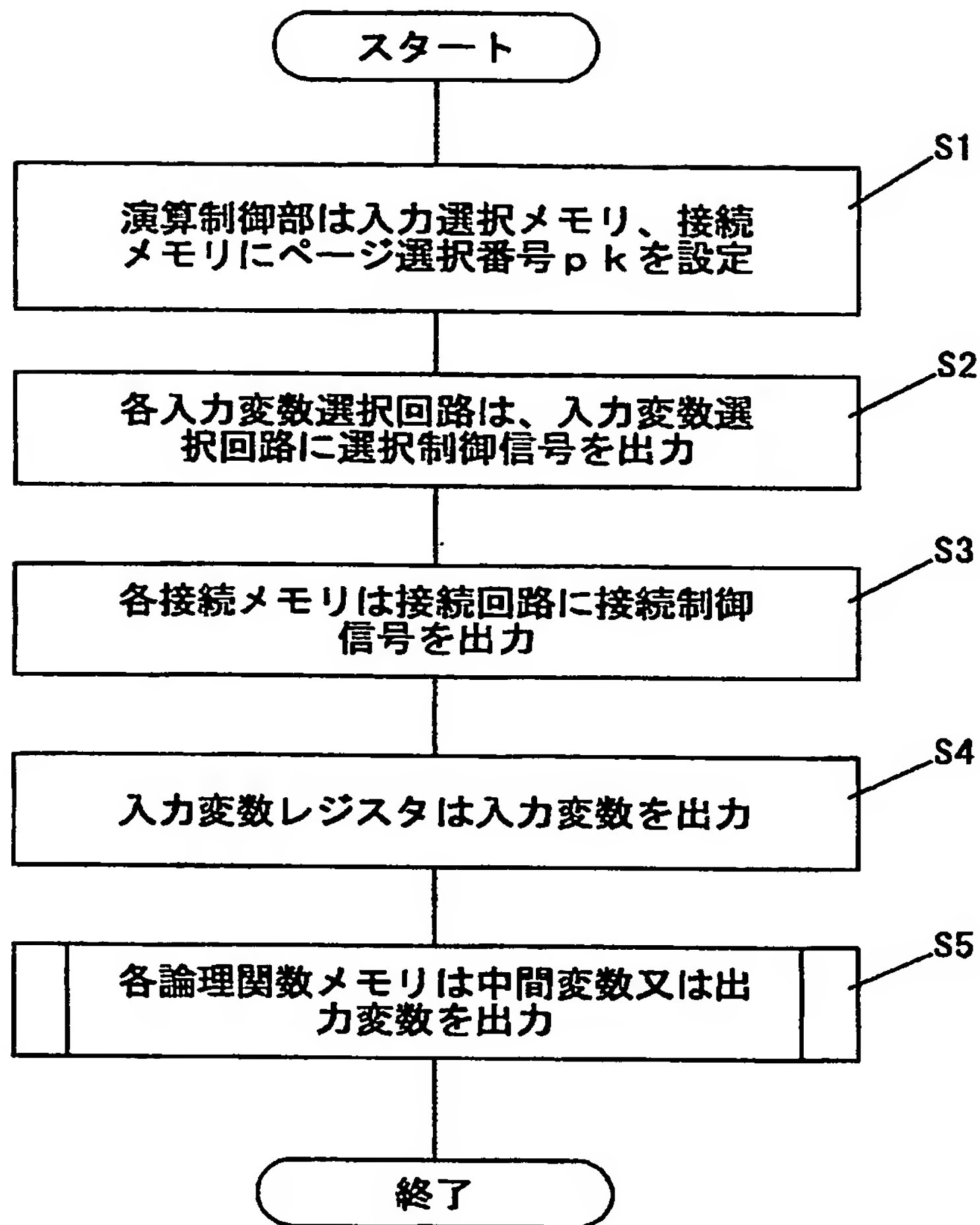
【図 4】



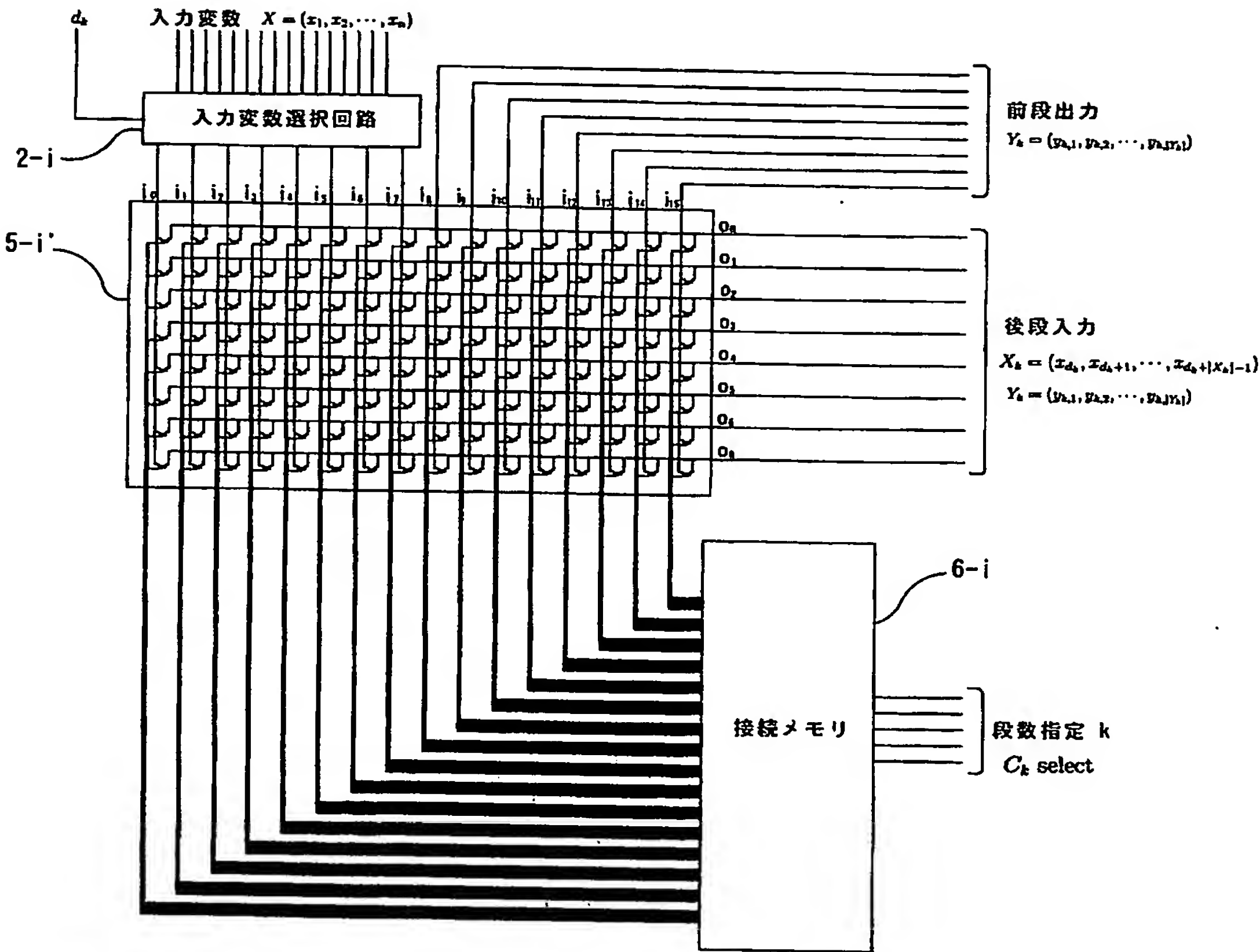
【図 5】



【図 6】

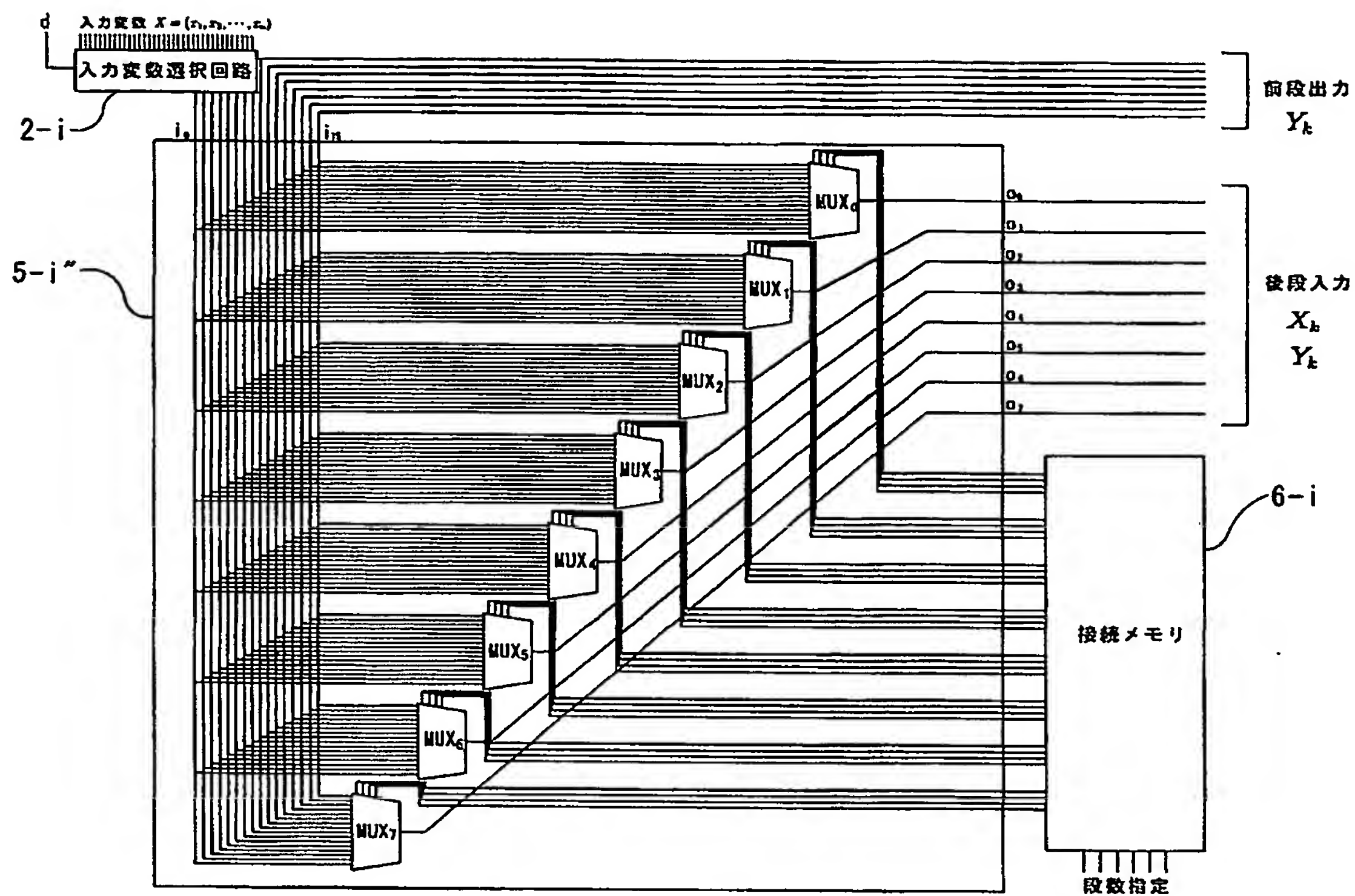


【図 7】

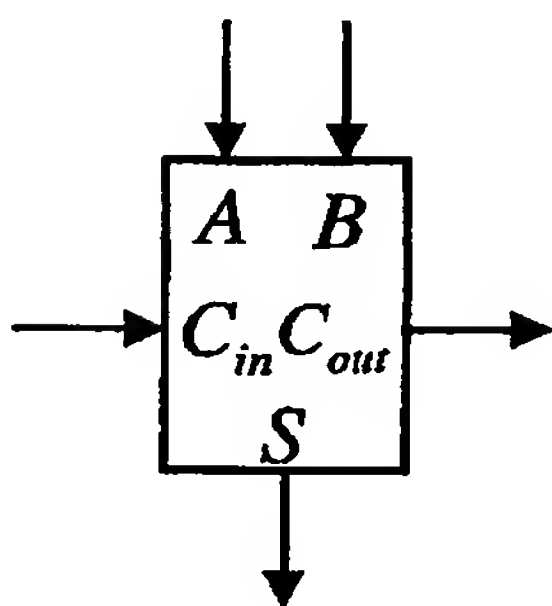




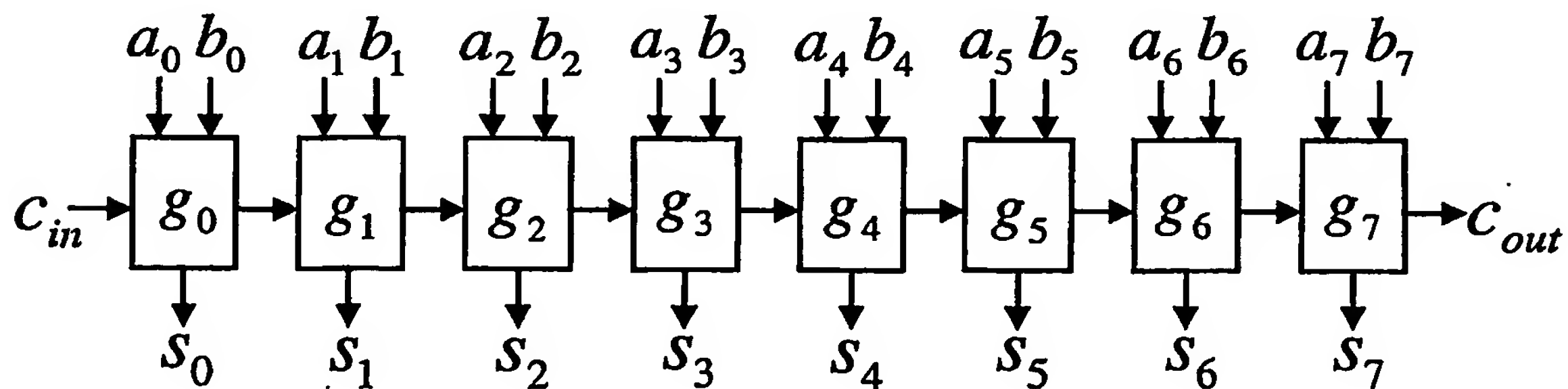
【図 8】



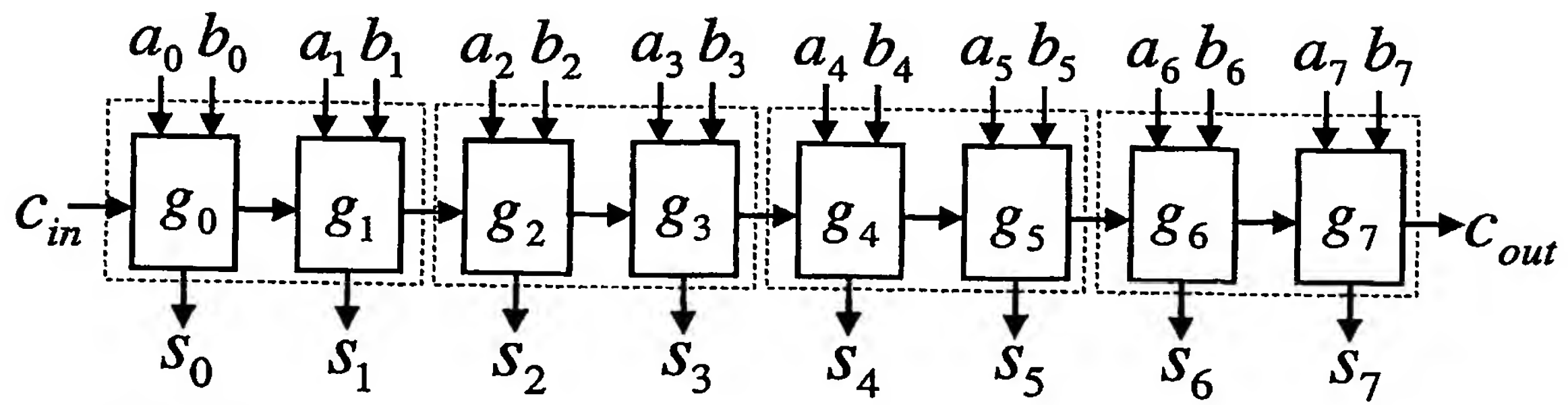
【図 9】



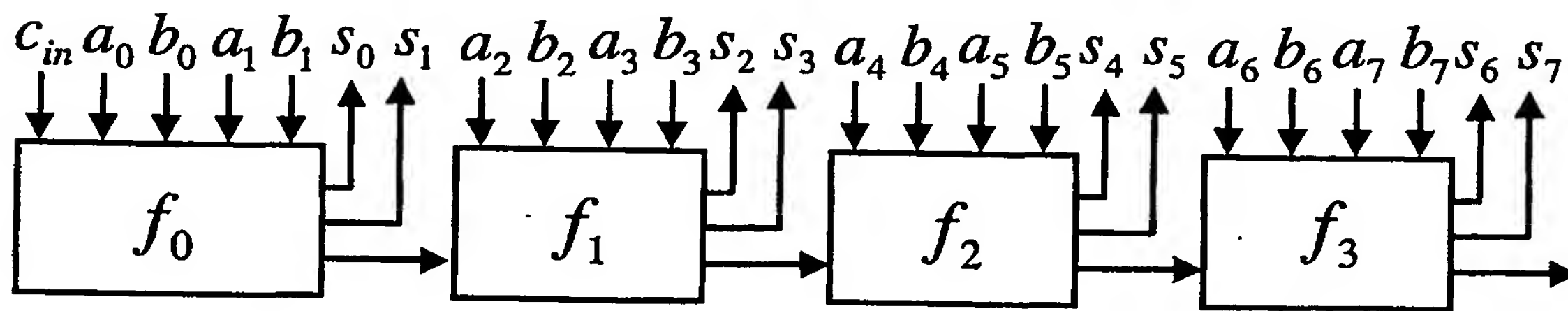
【図 10】



【図 1 1】



【図 1 2】



【図 13】

(a)

$c_{in}$	$a_1$	$a_0$	$b_1$	$b_0$	$c_{out}$	$s_1$	$s_0$	$c_{in}$	$a_1$	$a_0$	$b_1$	$b_0$	$c_{out}$	$s_1$	$s_0$
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0
0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	1
0	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0
0	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
0	0	1	1	0	0	1	1	1	0	1	1	0	1	0	0
0	0	1	1	1	1	0	0	1	0	1	1	1	1	0	1
0	1	0	0	0	0	1	0	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	1	1	0	0	1	0	0	1
0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0
0	1	0	1	1	1	0	1	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	1	1	1	1	0	0	1	0	0
0	1	1	0	1	1	0	0	1	1	1	0	1	1	0	1
0	1	1	0	1	1	0	0	1	1	1	1	0	1	0	0
0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0
0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1

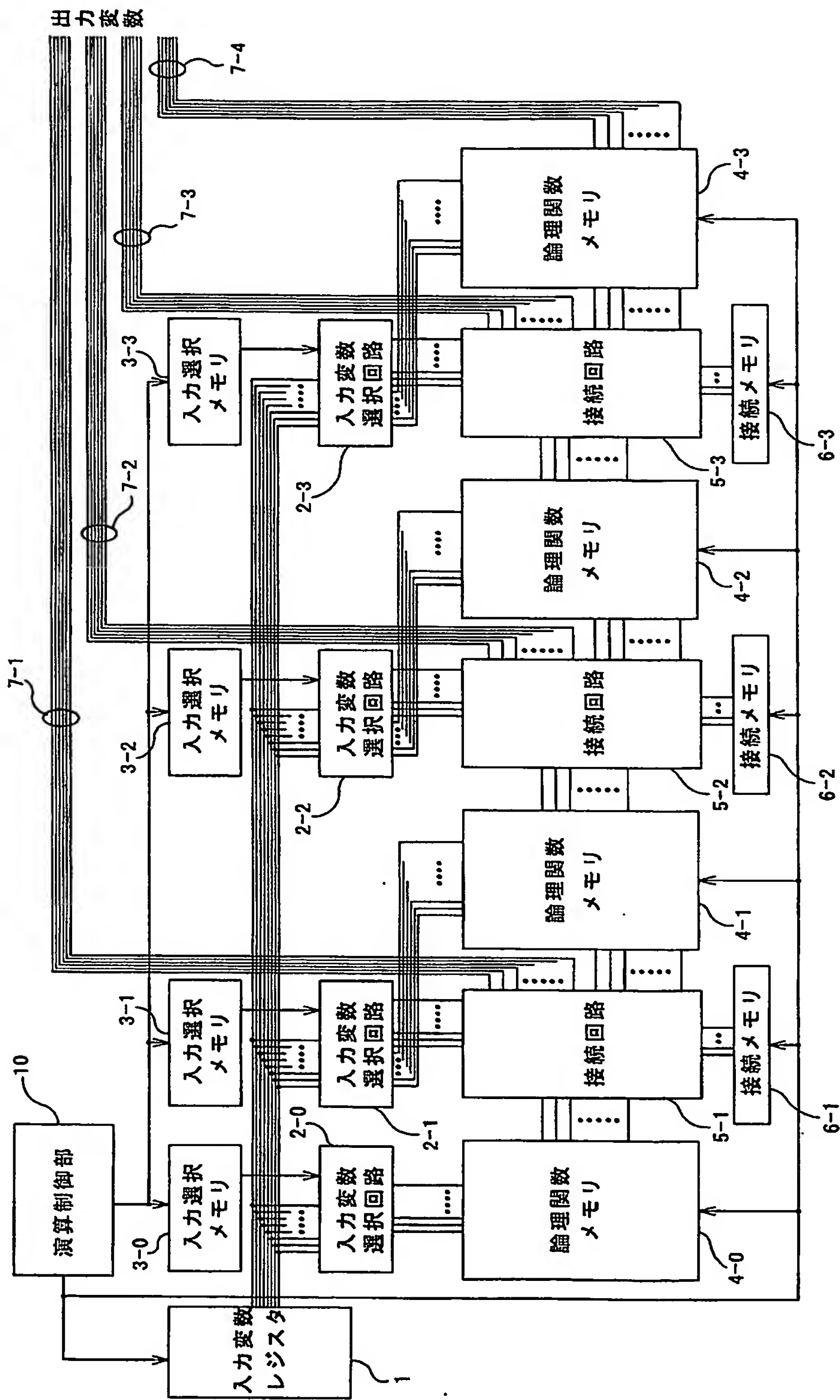
(b)

アドレス								出力値								アドレス								出力値								
$i_0$	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$	$i_6$	$i_7$	$o_7$	$o_6$	$o_5$	$o_4$	$o_3$	$o_2$	$o_1$	$o_0$	$i_0$	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$	$i_6$	$i_7$	$o_7$	$o_6$	$o_5$	$o_4$	$o_3$	$o_2$	$o_1$	$o_0$	
0	0	0	0	0	-	-	-	-	-	-	0	0	0	-	-	1	0	0	0	0	-	-	-	-	-	-	-	0	0	1	-	-
0	0	0	0	1	-	-	-	-	-	-	0	0	1	-	-	1	0	0	0	1	-	-	-	-	-	-	-	0	1	0	-	-
0	0	0	1	0	-	-	-	-	-	-	0	1	0	-	-	1	0	0	1	0	-	-	-	-	-	-	-	0	1	1	-	-
0	0	0	1	1	-	-	-	-	-	-	0	1	1	-	-	1	0	0	1	1	-	-	-	-	-	-	-	1	0	0	-	-
0	0	1	0	0	-	-	-	-	-	-	0	0	1	-	-	1	0	1	0	0	-	-	-	-	-	-	-	0	1	0	-	-
0	0	1	0	1	-	-	-	-	-	-	0	1	0	-	-	1	0	1	1	0	-	-	-	-	-	-	-	0	1	1	-	-
0	0	1	1	0	-	-	-	-	-	-	0	1	1	-	-	1	0	1	1	0	-	-	-	-	-	-	-	1	0	0	-	-
0	0	1	1	1	-	-	-	-	-	-	1	0	0	-	-	1	0	1	1	1	-	-	-	-	-	-	-	1	0	1	-	-
0	1	0	0	0	-	-	-	-	-	-	0	1	0	-	-	1	1	0	0	0	-	-	-	-	-	-	-	0	1	1	-	-
0	1	0	0	1	-	-	-	-	-	-	0	1	1	-	-	1	1	0	0	1	-	-	-	-	-	-	-	1	0	0	-	-
0	1	0	1	0	-	-	-	-	-	-	1	0	0	-	-	1	1	0	1	0	-	-	-	-	-	-	-	1	0	1	-	-
0	1	0	1	1	-	-	-	-	-	-	1	0	1	-	-	1	1	0	1	1	-	-	-	-	-	-	-	1	1	0	-	-
0	1	1	0	0	-	-	-	-	-	-	0	1	1	-	-	1	1	1	0	0	-	-	-	-	-	-	-	1	0	0	-	-
0	1	1	0	1	-	-	-	-	-	-	1	0	0	-	-	1	1	1	0	1	-	-	-	-	-	-	-	1	0	1	-	-
0	1	1	1	0	-	-	-	-	-	-	1	0	1	-	-	1	1	1	1	0	-	-	-	-	-	-	-	1	1	0	-	-
0	1	1	1	1	-	-	-	-	-	-	1	1	0	-	-	1	1	1	1	1	-	-	-	-	-	-	-	1	1	1	-	-

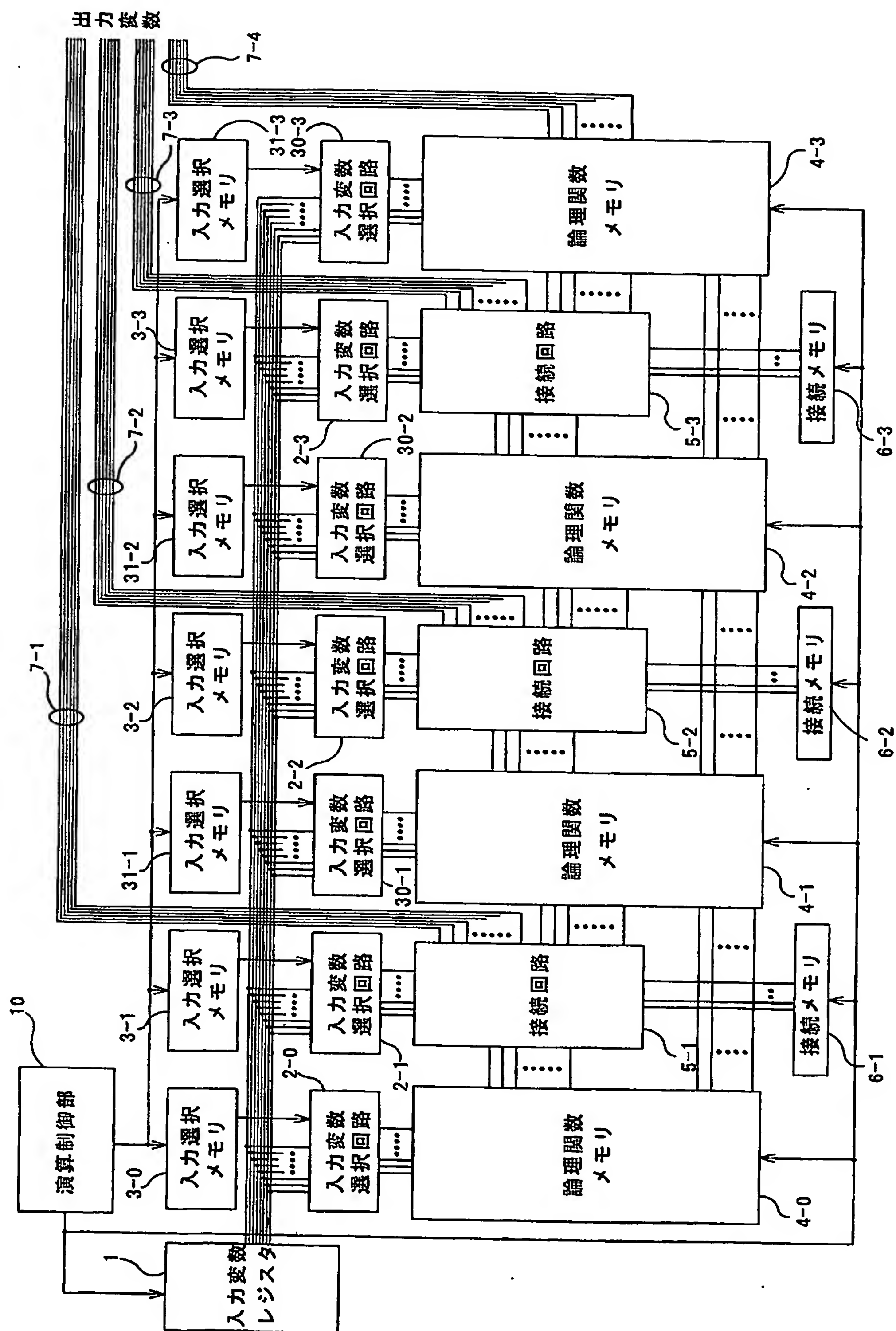
(c)

アドレス								出力値								アドレス								出力値								
i <sub>0</sub>	i <sub>1</sub>	i <sub>2</sub>	i <sub>3</sub>	i <sub>4</sub>	i <sub>5</sub>	i <sub>6</sub>	i <sub>7</sub>	o <sub>7</sub>	o <sub>6</sub>	o <sub>5</sub>	o <sub>4</sub>	o <sub>3</sub>	o <sub>2</sub>	o <sub>1</sub>	o <sub>0</sub>	i <sub>0</sub>	i <sub>1</sub>	i <sub>2</sub>	i <sub>3</sub>	i <sub>4</sub>	i <sub>5</sub>	i <sub>6</sub>	i <sub>7</sub>	o <sub>7</sub>	o <sub>6</sub>	o <sub>5</sub>	o <sub>4</sub>	o <sub>3</sub>	o <sub>2</sub>	o <sub>1</sub>	o <sub>0</sub>	
0	-	-	-	0	0	0	0	-	-	-	0	0	0	-	-	1	-	-	-	0	0	0	0	-	-	-	-	0	0	1	-	-
0	-	-	-	0	0	0	1	-	-	-	0	0	1	-	-	1	-	-	-	0	0	0	1	-	-	-	-	0	1	0	-	-
0	-	-	-	0	0	1	0	-	-	-	0	1	0	-	-	1	-	-	-	0	0	1	0	-	-	-	-	0	1	1	-	-
0	-	-	-	0	0	1	1	-	-	-	0	1	1	-	-	1	-	-	-	0	0	1	1	-	-	-	-	1	0	0	-	-
0	-	-	-	0	1	0	0	-	-	-	0	0	1	-	-	1	-	-	-	0	1	0	0	-	-	-	-	0	1	0	-	-
0	-	-	-	0	1	0	1	-	-	-	0	1	0	-	-	1	-	-	-	0	1	0	1	-	-	-	-	0	1	1	-	-
0	-	-	-	0	1	1	0	-	-	-	0	1	1	-	-	1	-	-	-	0	1	1	0	-	-	-	-	1	0	0	-	-
0	-	-	-	0	1	1	1	-	-	-	1	0	0	-	-	1	-	-	-	1	0	1	1	-	-	-	-	1	0	1	-	-
0	-	-	-	1	0	0	0	-	-	-	0	1	0	-	-	1	-	-	-	1	0	0	0	-	-	-	-	1	0	1	-	-
0	-	-	-	1	0	0	1	-	-	-	0	1	1	-	-	1	-	-	-	1	0	1	1	-	-	-	-	1	0	1	-	-
0	-	-	-	1	0	1	0	-	-	-	1	0	0	-	-	1	-	-	-	1	0	0	1	-	-	-	-	1	0	0	-	-
0	-	-	-	1	0	1	1	-	-	-	1	0	1	-	-	1	-	-	-	1	0	1	0	-	-	-	-	1	0	1	-	-
0	-	-	-	1	1	0	0	-	-	-	0	1	1	-	-	1	-	-	-	1	1	0	0	-	-	-	-	1	0	0	-	-
0	-	-	-	1	1	0	1	-	-	-	1	0	0	-	-	1	-	-	-	1	1	0	1	-	-	-	-	1	0	1	-	-
0	-	-	-	1	1	1	0	-	-	-	1	0	1	-	-	1	-	-	-	1	1	1	0	-	-	-	-	1	1	0	-	-
0	-	-	-	1	1	1	1	-	-	-	1	1	0	-	-	1	-	-	-	1	1	1	1	-	-	-	-	1	1	1	-	-

【図14】

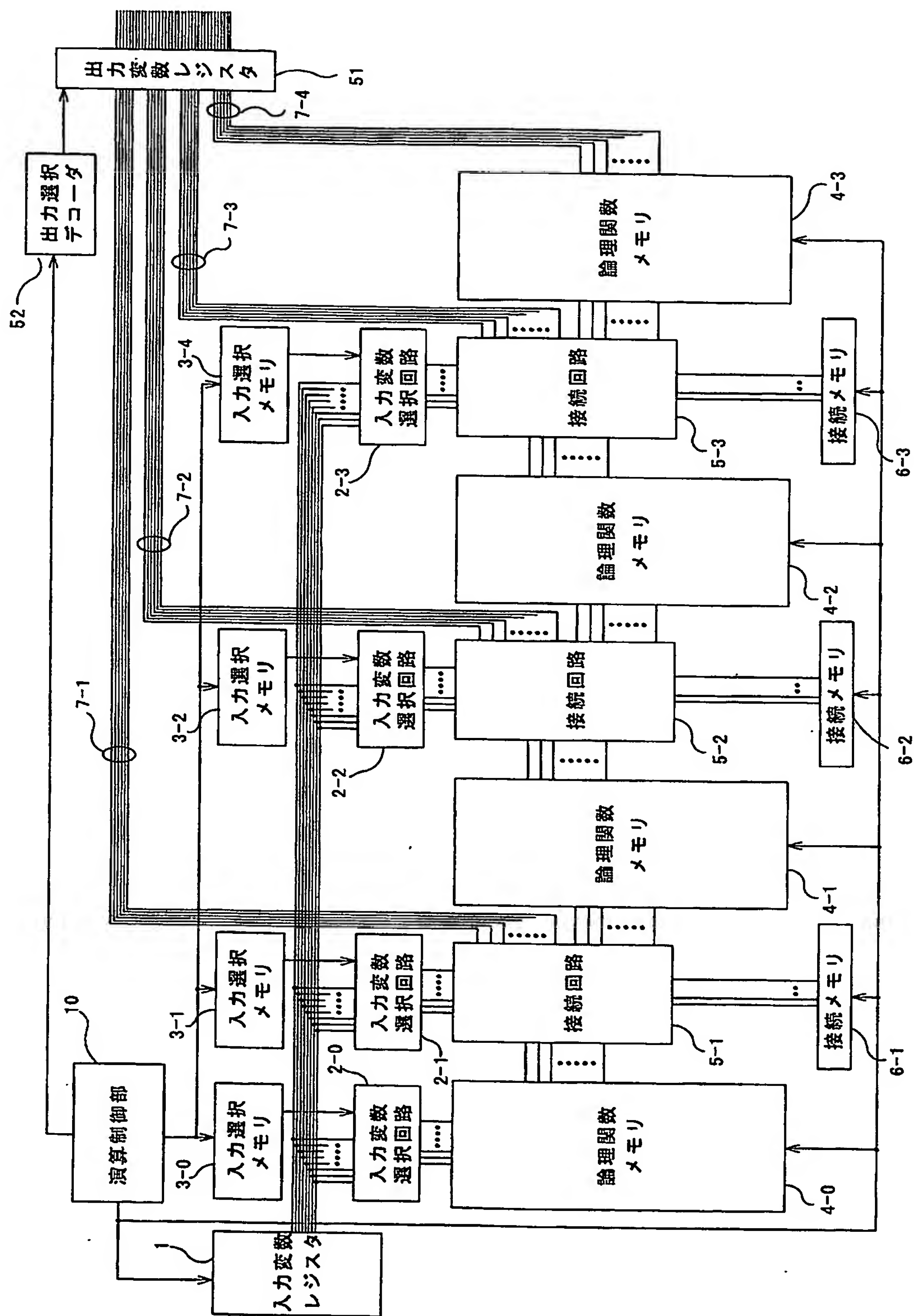


【圖 15】

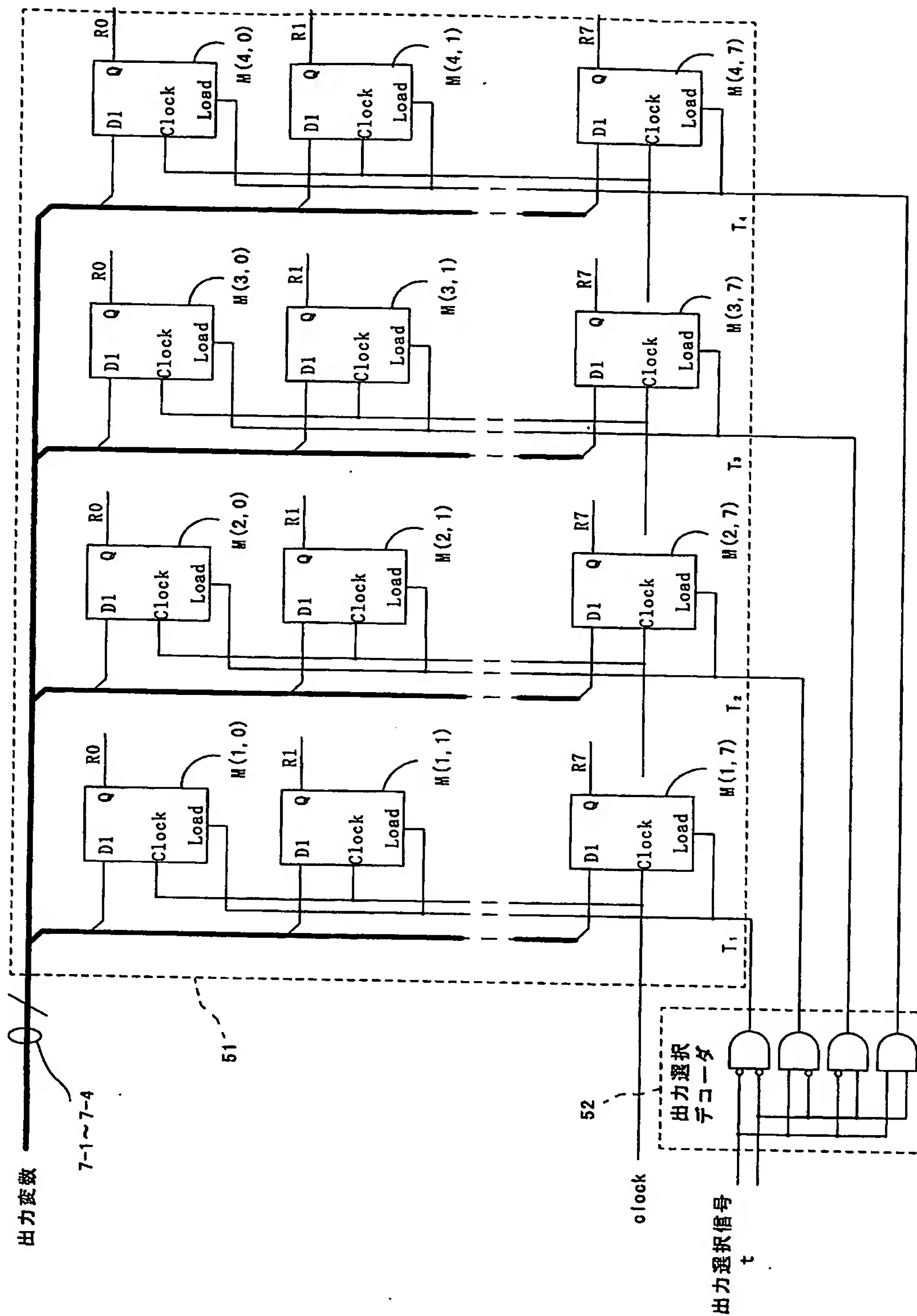




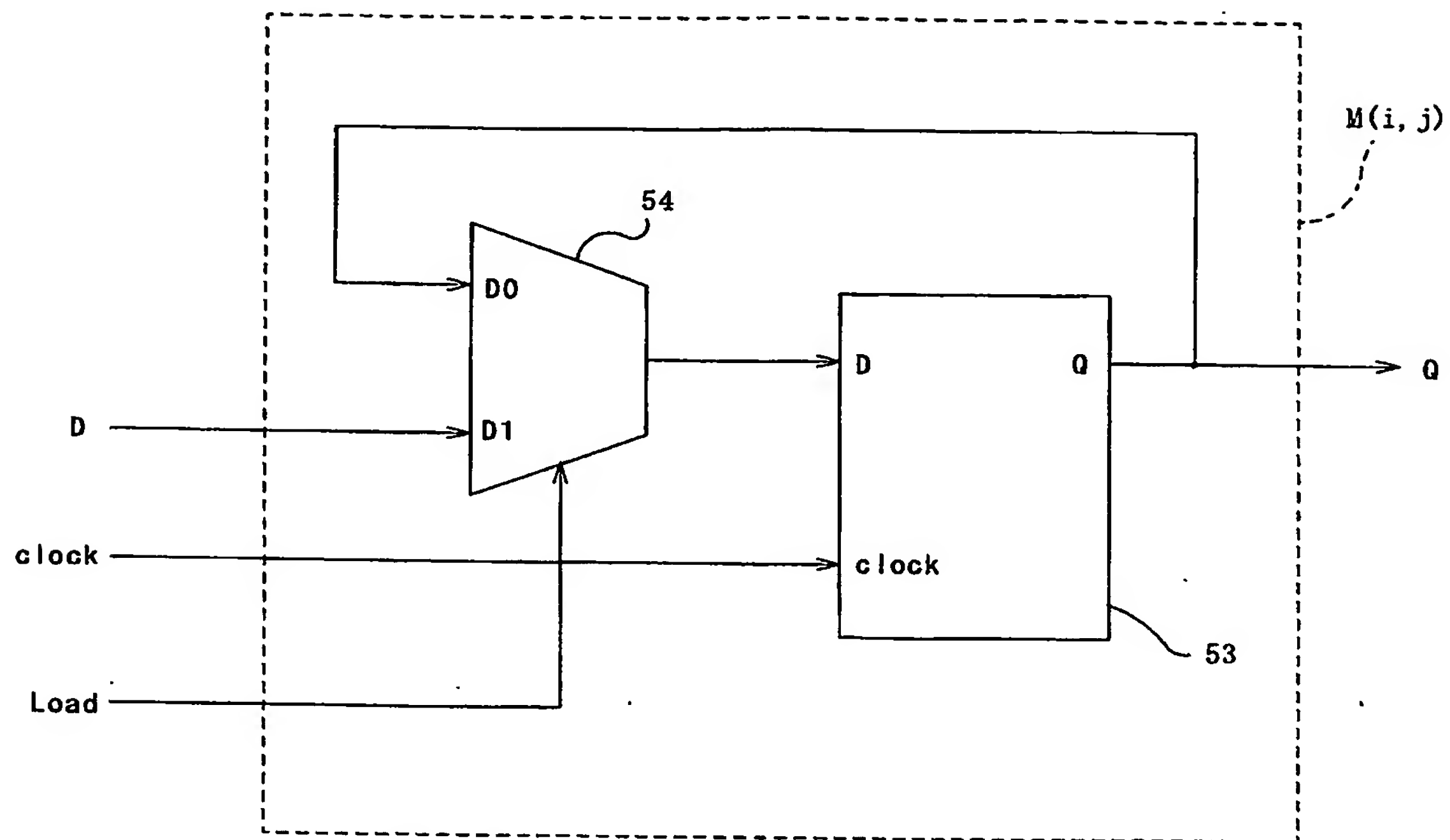
【図16】



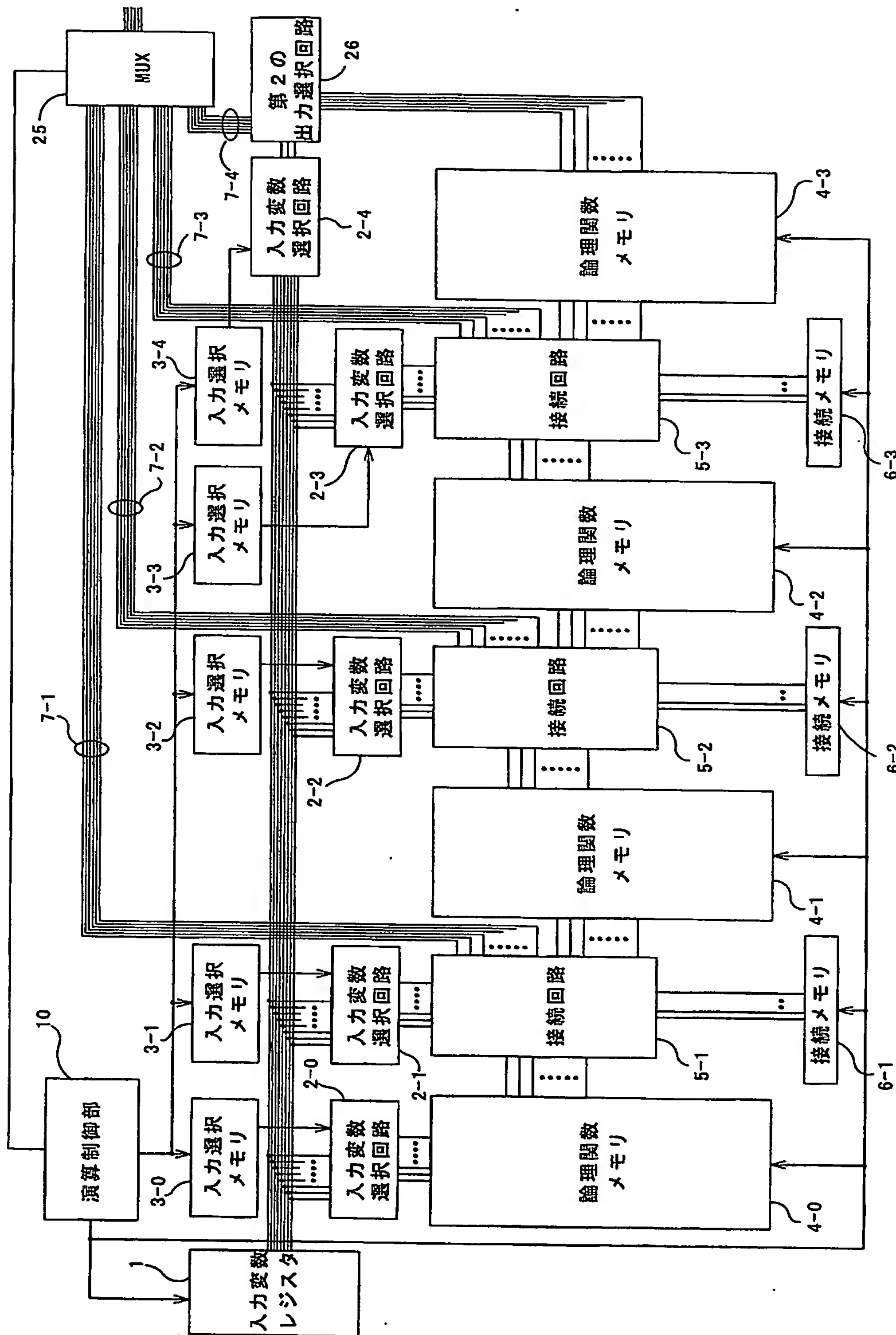
【図 17】



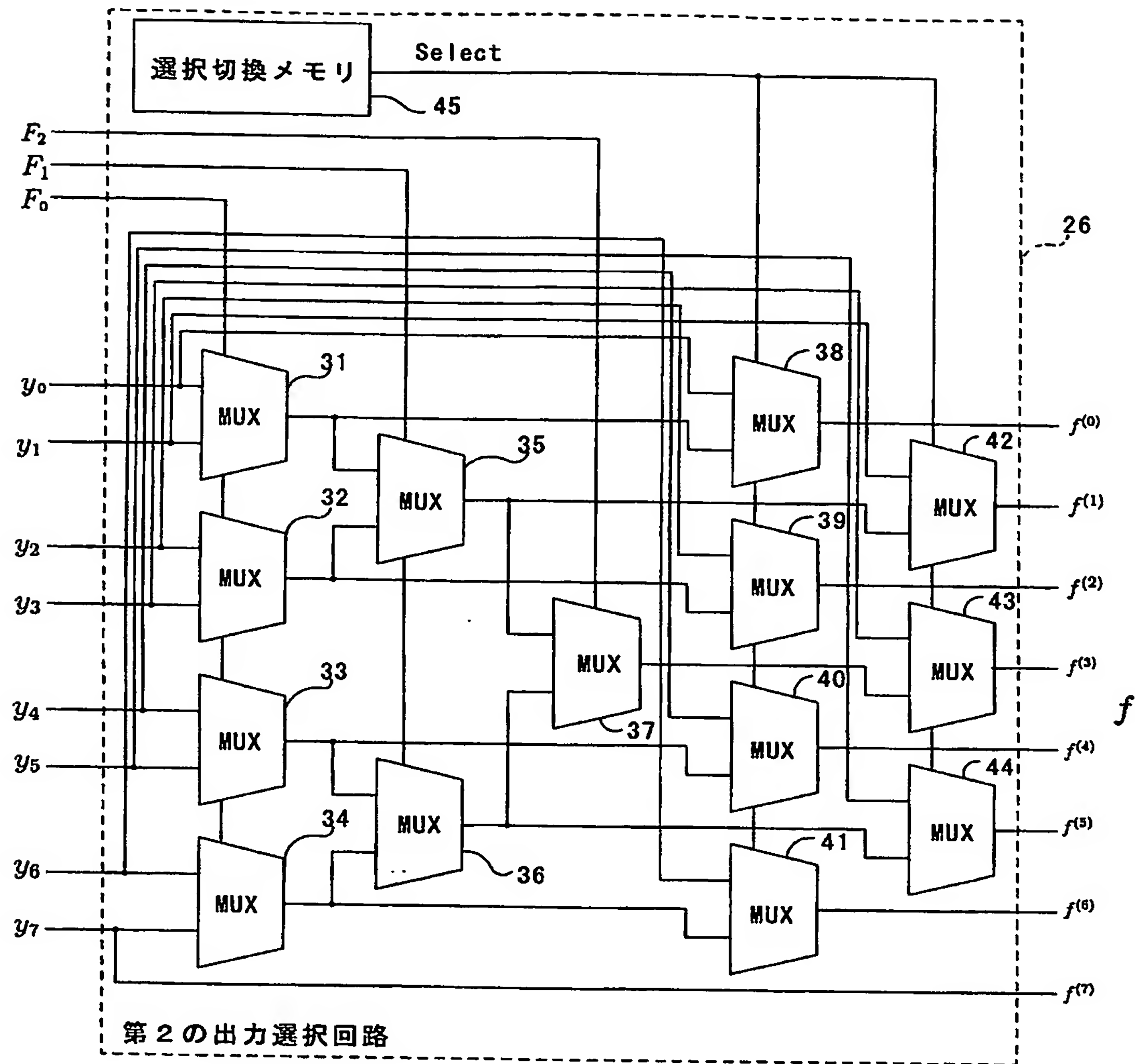
【図 18】



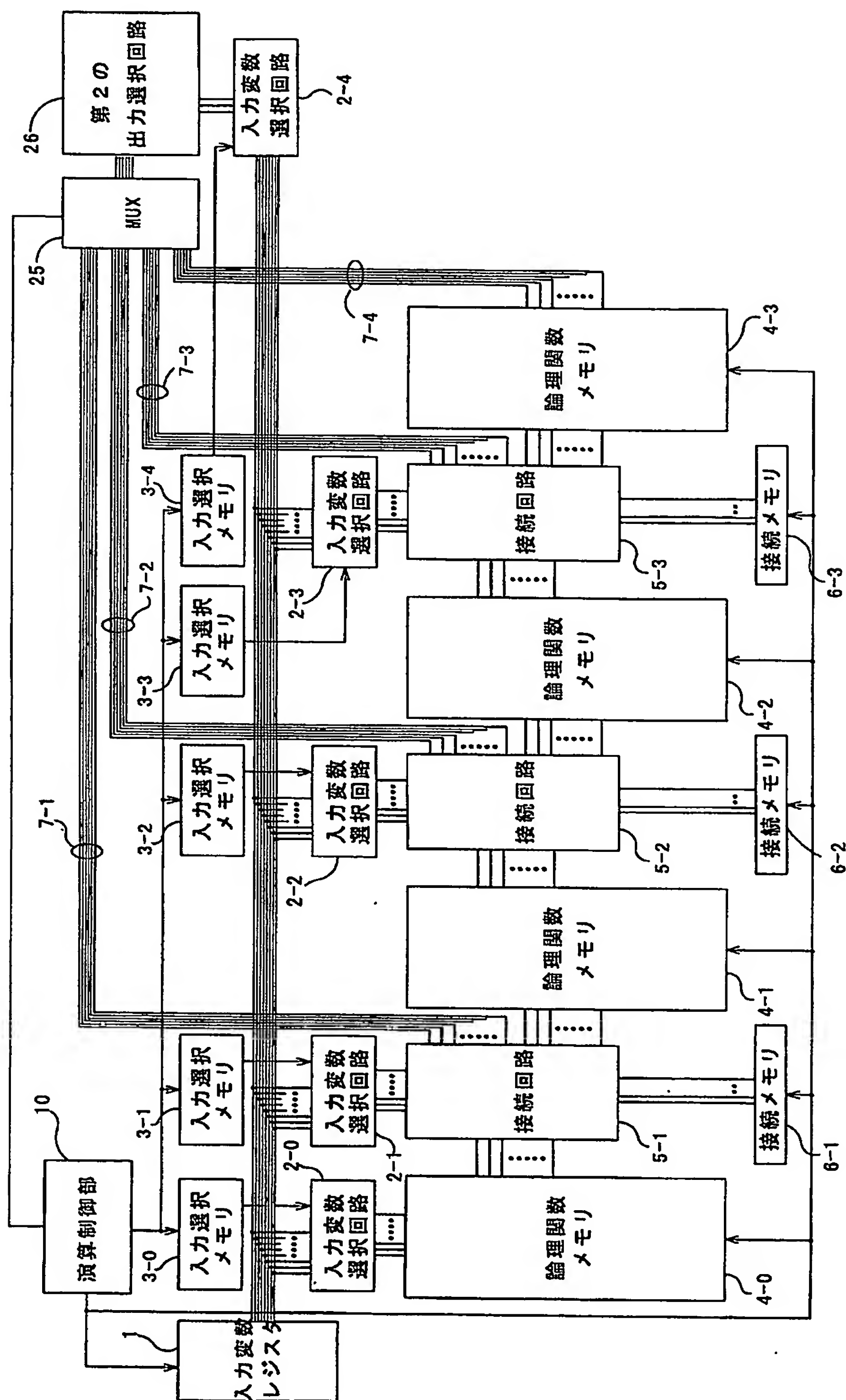
【図 19】



【図20】

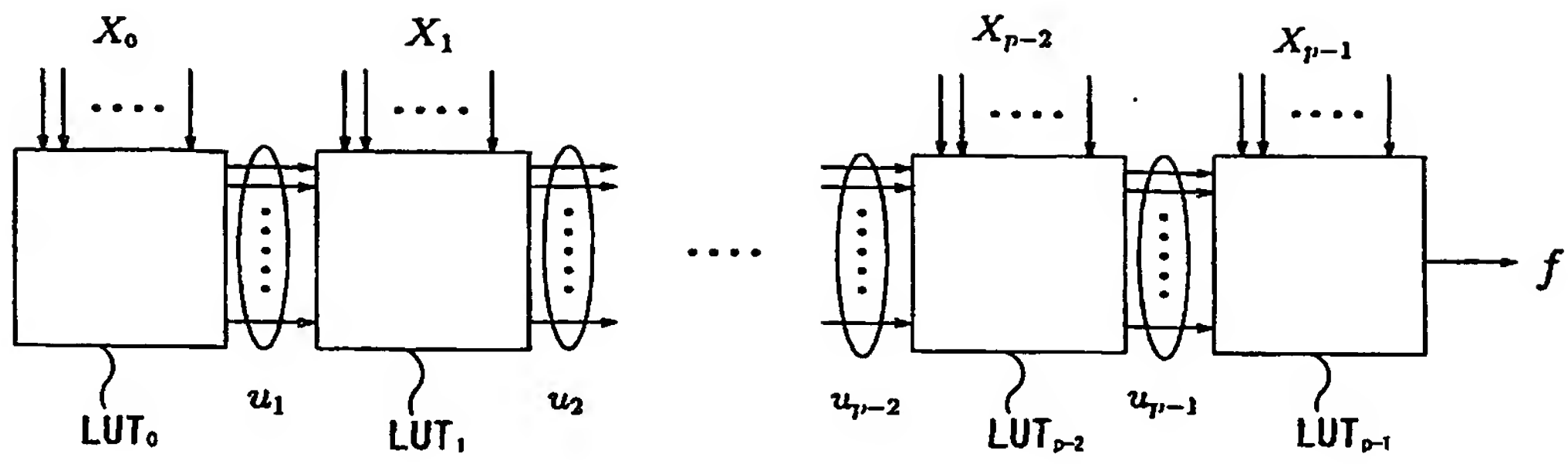


【図 21】





【図 22】



【書類名】 要約書

【要約】

【課題】 複数の目的論理関数についての論理回路を設計する場合であっても、レイアウト面積を小さくすることが可能なルックアップテーブル・カスケード論理回路を提供する。

【解決手段】 ルックアップテーブル・カスケード論理回路において、各分解関数をルックアップテーブルとして記憶する、直列に配列された複数の論理関数メモリ 4 と、前段の論理関数メモリ 4 の複数個の出力及び入力変数の入力のうち、後段の論理関数メモリ 4 の複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶する接続メモリ 6 と、接続メモリ 6 より出力される接続情報に従い、前段の論理関数メモリ 4 の出力又は前記入力変数の入力と後段の論理関数メモリ 4 の入力とを選択的に接続する接続回路 5 とを備えた。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 9 3 9 2 2
受付番号	5 0 3 0 0 5 2 7 3 1 7
書類名	特許願
担当官	第八担当上席 0 0 9 7
作成日	平成 1 5 年 4 月 1 日

< 認定情報・付加情報 >  
【提出日】 平成15年 3月31日

【書類名】 出願人名義変更届  
【あて先】 特許庁長官殿  
【事件の表示】  
【出願番号】 特願2003- 93922  
【承継人】  
【識別番号】 802000031  
【氏名又は名称】 財団法人北九州産業学術推進機構  
【代表者】 有馬 朗人  
【承継人代理人】  
【識別番号】 100121371  
【弁理士】  
【氏名又は名称】 石田 和人  
【電話番号】 093-695-3113  
【手数料の表示】  
【予納台帳番号】 222495  
【納付金額】 4,200円  
【提出物件の目録】  
【物件名】 承継人であることを証する書面 1  
【援用の表示】 平成15年12月12日提出の手續補足書により補足する。  
【物件名】 委任状 1  
【援用の表示】 平成15年12月12日提出のものを援用する。

## 認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 9 3 9 2 2
受付番号	5 0 3 0 2 0 5 3 2 7 1
書類名	出願人名義変更届
担当官	古田島 千恵子 7 2 8 8
作成日	平成 1 6 年 3 月 2 5 日

## &lt; 認定情報・付加情報 &gt;

【提出日】	平成15年12月12日
-------	-------------

【承継人】	
-------	--

【識別番号】	802000031
--------	-----------

【住所又は居所】	福岡県北九州市若松区ひびきの2番1号
----------	--------------------

【氏名又は名称】	財団法人北九州産業学術推進機構
----------	-----------------

【承継人代理人】	申請人
----------	-----

【識別番号】	100121371
--------	-----------

【住所又は居所】	福岡県北九州市若松区ひびきの2番1号 北九州 学術研究都市産学連携センターT-302 石田 特許事務所内
----------	--

【氏名又は名称】	石田 和人
----------	-------

特願 2003-093922

出願人履歴情報

識別番号 [502428467]

1. 変更年月日 2002年11月26日  
[変更理由] 新規登録  
住 所 福岡県福岡市中央区今川1丁目10-43-403  
氏 名 笹尾 勤
2. 変更年月日 2003年 5月25日  
[変更理由] 住所変更  
住 所 福岡県福岡市中央区鳥飼3丁目15-17  
氏 名 笹尾 勤



特願 2 0 0 3 - 0 9 3 9 2 2

出 願 人 履 歴 情 報

識別番号 [ 8 0 2 0 0 0 0 3 1 ]

1. 変更年月日	2 0 0 2 年 4 月 1 9 日
[変更理由]	新規登録
住 所	福岡県北九州市若松区ひびきの 2 番 1 号
氏 名	財団法人北九州産業学術推進機構